UniPlot Manual

Release R2015

Uniplot Software GmbH

December 02, 2015

Contents

1 Installation of UniPlot **1**
- 1.1 System Requirements . . . 1
- 1.2 Installation . . . 1
- 1.3 UniPlot Directory . . . 2
- 1.4 Command Line Options . . . 2
- 1.5 Using your own setup program . . . 2
- 1.6 Network Installation . . . 3
- 1.7 Deinstallation . . . 3

2 Documentation, Support **5**

3 UniPlot Essentials **7**

4 A Simple Example **13**
- 4.1 Creating a Document . . . 13
- 4.2 Importing Data into a Diagram . . . 14
- 4.3 Editing Isolines . . . 17
- 4.4 3D Dataset Configuration . . . 19
- 4.5 Diagram and Axes Titles . . . 21
- 4.6 Editing Diagram Configuration . . . 22
- 4.7 Importing a 2D Dataset . . . 23

5 UniPlot Objects **25**
- 5.1 UniPlot Documents . . . 25
- 5.2 2D Diagram . . . 27
- 5.3 Stacked diagrams with multiple y axes . . . 30
- 5.4 3D Diagram . . . 33
- 5.5 Waterfall Diagram . . . 33
- 5.6 Dataset Objects . . . 34
- 5.7 Drawing Objects . . . 34
- 5.8 Formula and Latex Text . . . 36
- 5.9 Text Objects . . . 36
- 5.10 Text Placeholder . . . 42
- 5.11 Field Functions . . . 44

5.12 Information/Legend Boxes . . . 47
5.13 Table Objects . . . 49

6 The Clipboard **53**
6.1 Insert Objects into UniPlot . . . 53
6.2 Editing an Embedded Object . . . 53
6.3 Copying Drawing Objects . . . 54
6.4 Copying Datasets . . . 54
6.5 Copy Diagrams . . . 54
6.6 Copying Pages . . . 54

7 Examples **55**
7.1 2D Dataset Attributes . . . 55
7.2 Diagram with 3 Overlapping y-axes . . . 56
7.3 2D Dataset - Derivation and Integration . . . 57
7.4 Diagram with 2 Curves and Color Filling . . . 58
7.5 Plot a Function f(x) . . . 59
7.6 Contour Diagram with Color Gradient and Legend . . . 60
7.7 Contour and 3D Diagram . . . 61
7.8 Plot a Function f(x,y) . . . 62
7.9 Contour Cross-section . . . 63
7.10 Cross-section along a 2D Curve . . . 64

8 Importing Data **67**
8.1 Load a dataset . . . 68
8.2 Format Description for CSV, Text and Excel Files . . . 70
8.3 UTX Data File Format . . . 71
8.4 Import 3D Matrix Data . . . 79
8.5 Definition of the Full Load Line (WOT) . . . 80
8.6 Arbitrary Data Hull . . . 82
8.7 Save Hull . . . 83
8.8 Load Hull . . . 83
8.9 Waterfall Data Import . . . 83

9 Overview Automation **87**
9.1 Automation from the user's point of view . . . 87
9.2 Creating an automation . . . 88

10 Licenses **95**
10.1 UniPlot . . . 95
10.2 netCDF . . . 95
10.3 Scintilla . . . 96
10.4 VTK . . . 97
10.5 PCRE . . . 97
10.6 zlib . . . 99
10.7 LZF . . . 99
10.8 expat . . . 100
10.9 qhull . . . 100

10.10 jpeg . . . 101
10.11 png . . . 101
10.12 cxImage . . . 103
10.13 nullsoft Installer . . . 103
10.14 libffi . . . 104
10.15 HDF5 . . . 104
10.16 kazlib . . . 106
10.17 upfirdn (Polyphase FIR Resampling) . . . 106
10.18 LZ4 . . . 107
10.19 libxml2 . . . 107
10.20 Color Brewer . . . 108
10.21 cURL Library (http://curl.haxx.se) . . . 108
10.22 OpenSSL . . . 109
10.23 libssh2 . . . 111

Index **113**

CHAPTER 1

Installation of UniPlot

1.1 System Requirements

UniPlot can be used with the following operating systems:

- Windows XP
- Windows Vista
- Windows 7
- Windows 8

1.2 Installation

- Place the CD-ROM in your CD-ROM drive. The setup program should start automatically after a few seconds.
- If the setup program does not start automatically, please start the program manually by selecting **Start=>Run** and choose the setup program in the `\setup` folder or download the current version from

  http://www.uniplot.de/download.html

- Please make sure to have a valid license key handy. The key is needed during setup.
- Two setup programs are available:

  ```
  setup-uniplot-R2013_3-win64.exe
  setup-uniplot-R2013_3-win32.exe
  ```

  In 64 bit Windows, please install the 64 bit version of UniPlot (`setup-uniplot-R2013_3-win64.exe`), otherwise use the 32 bit version.

  In a 64 bit Windows you can install either the 64 bit version or the 32 bit version of UniPlot. In a 32 bit Windows only the 32 bit UniPlot version can be started.

- All program and example files will be copied onto your hard drive into the `c:\Program Files (x86)\UniPlot Software\R2013-win64` directory or a directory of your choice.

- After UniPlot has been successfully installed, start UniPlot with a double click on the program icon.

Please contact us if you have any problem installing UniPlot or starting the program.

1.3 UniPlot Directory

The current 64-Bit UniPlot version is normally installed in the following directory:

```
c:\Program Files (x86)\UniPlot Software\R2013-win64
```

The 32-Bit version is installed in:

```
c:\Program Files (x86)\UniPlot Software\R2013-win32
```

GetRootDirectory returns the UniPlot directory.

1.4 Command Line Options

In the command line of the setup program you can assign the license key and the installation directory.

Example:

```
setup-uniplot-R2013_13-win64.exe /S /LIC=NNN /ALLUSERS /D=C:\Program Files (x86)\UniPlot So
```

Parameter	Meaning
/S	Silent. The setup program will not display any dialogs.
/Lic=NNN	License key. NNN must be replaced by a valid license key.
/ALLUSERS	Icons for all users will be created. Default is only active user.
/D=SSS	/D sets the default installation directory. It must be the last parameter used in the command line and must not contain any quotes, even if the path contains spaces. Only absolute paths are supported.

1.5 Using your own setup program

To setup UniPlot on a computer you can simply copy the complete UniPlot folder to a new location on the target computer, for example `C:\Program Files\UniPlot Software\R2013` and start `uniplot.exe`. When UniPlot is started from the new location, three UniPlot Icons (in the Start menu, on the Desktop and in the Quick Launch menu bar) are created and all the necessary registry settings are saved in the system registry for the current user. You can find all keys that will be saved in the registry in the `script\uniplot\post-install.ic` file.

1.6 Network Installation

If you own multiple UniPlot licenses and you can ensure that only the correct number of licenses is used, you can install UniPlot on a server.

When UniPlot is started from a client computer, three UniPlot Icons (in the Start menu, on the Desktop and in the Quick Launch menu bar) are created and all the necessary registry settings are saved in the system registry for the current user.

1.7 Deinstallation

To uninstall UniPlot, open Programs and Features by clicking the **Start** button, clicking Control Panel, clicking Programs, and then clicking Programs and Features. Choose **UniPlot R2013**.

Alternatively, you can run the following command:

```
c:\Program Files(x86)\UniPlot Software\R2013\uninstall.exe /S
```

The uninstall program does not remove the UniPlot user directory:

```
C:\Users\<USER-NAME>\Documents\UniPlot\
```

The uninstall program does not remove the UniPlot registry settings in:

```
HKEY_CURRENT_USER\Software\RSB
```

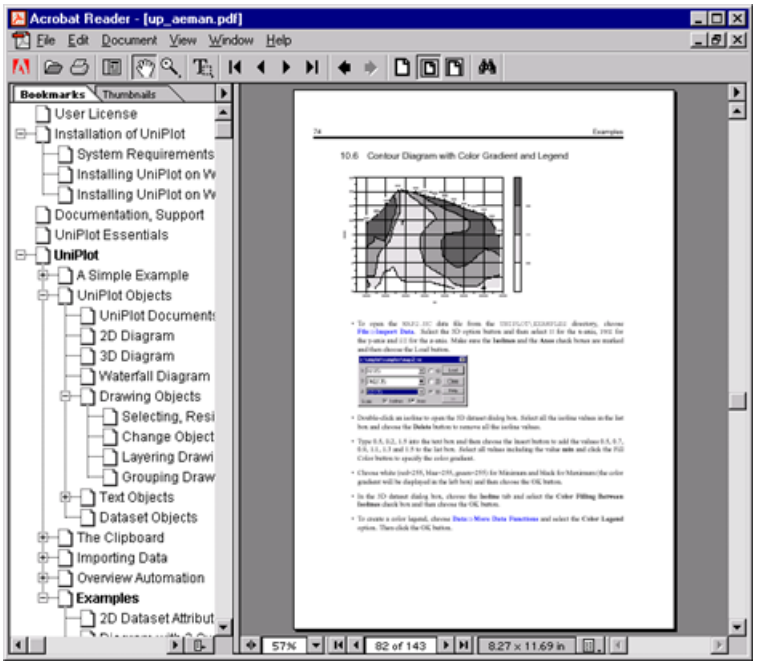
CHAPTER 2

Documentation, Support

The UniPlot User Manual contains 2 parts: UniPlot and UniScript. UniScript is the built in script language that can be used to extend the functionality of UniPlot.

UniPlot documentation is also available in PDF-Format (see http://www.uniplot.de/documents/de/download.html)

It can be viewed and printed with Acrobat-Reader (Acrobat-Reader of Adobe Systems Incorporated http://www.adobe.com).



Please take a moment to read the release notes which contain important information that is not found in the manual.

The manual does not contain UniPlot's complete documentation. The documentation contains more parts which are found in online Help system.

Updates:

UniPlot and this user manual are updated regularly. Please take a look from time to time at our UniPlot Homepage (http://www.uniplot.de or http://www.uniplot.com).

Support:

Updates can be downloaded as you have purchased UniPlot: Please check from time to time our home page (http://www.uniplot.de or http://www.uniplot.com).

If you would like to be informed about updates and other UniPlot news please subscribe our RSS-Feed http://www.uniplot.com/news.html. (More about RSS in Wikipedia: http://en.wikipedia.org/wiki/RSS)

If for some reason this manual or the online Help System do not provide the answer you are looking for, please contact us. We can be reached by phone between 9 - 17, or you can write us a letter, fax or E-Mail:

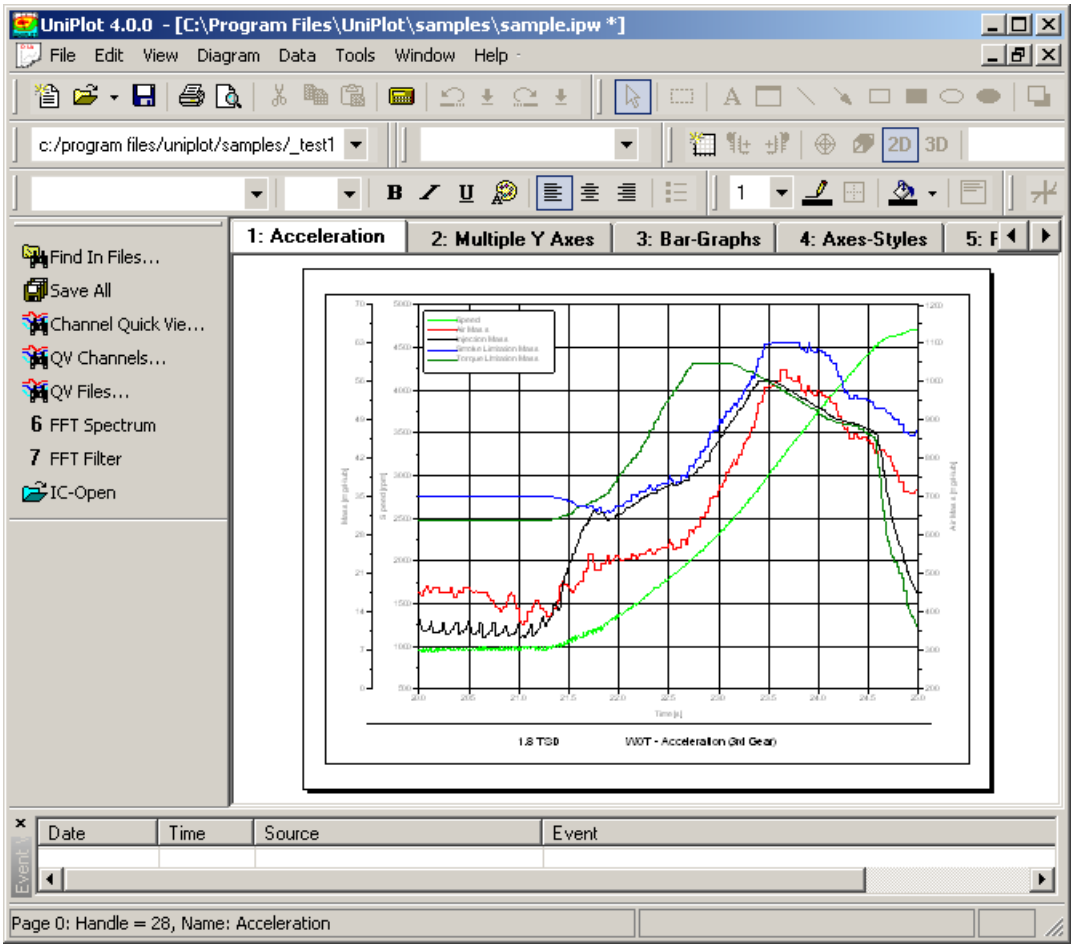
Uniplot Software GmbH
Am Seekanal 16
D-15834 Rangsdorf
Germany

Tel.: +49 33708/73754
Fax: +49 33708/73755

support@uniplot.de
Internet: http://www.uniplot.de
or http://www.uniplot.com

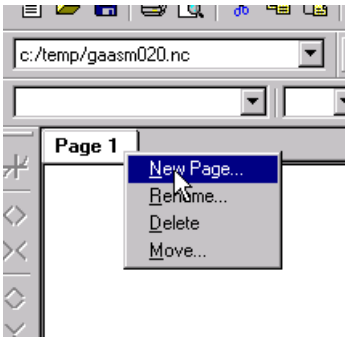
CHAPTER 3

UniPlot Essentials



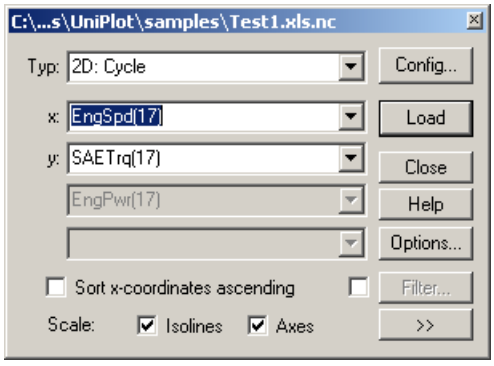
- Example plots are found in the `UniPlot\Samples` directory and can be opened with the **Open** command in the **File** menu (*fileopen*) or by clicking .
- To select a diagram, a dataset or a drawing object, click the object with the mouse. To edit the object settings, double-click an object (axis, axis label, axis title, text, dataset, diagram, etc.) to open the configuration dialog box.
- The dataset configuration dialog box can also be opened with the **1D and 2D Dataset Configuration** and **3D Dataset Configuration** buttons.

- To move a diagram, position the cursor on the frame (but not on a handle) and then drag it. To resize a diagram or a drawing object, drag any handle.
- When you drag the handle of an object, the object is pulled into alignment with the gridline or the nearest intersection of gridlines (Snap-To-Grid effect). To turn off this effect, hold down the `Ctrl` key while you drag the object. To change the spacing between the gridlines, choose the **Options** dialog box in the **Tools** menu (*toolsoptions*).
- Drawing objects and diagrams can also be moved by using the arrow keys on the keyboard. While pressing the `Shift` key, the width and height of the diagram or drawing object can be changed. To turn off the Snap-To-Grid-Effect press the `Ctrl` key.
- Drawing objects and diagrams can also be dragged by using the arrow keys. To change the size of an object, hold down the `Shift` key while using an arrow key. To switch off the Snap-To-Grid effect hold down the `Ctrl` key.
- To create a new document choose *file-new* or the button.
- To create a new document, click the button on the toolbar. A diagram can be added to the document by clicking the button and then dragging the mouse to create the diagram. Diagrams with multiple axes can be created by adding multiple diagrams. The axis position can be set in the *diagram-diagram-settings* dialog box. To move an axis from its standard position, open the *diagram-layout* dialog box ( ).
- To add a new page to the active document, position the cursor on one of the page tabs and click the right mouse key or choose *editpagenew-page*.



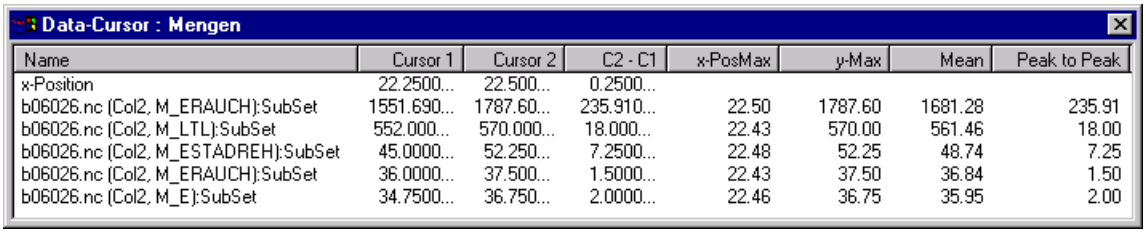
- To create a drawing object, choose one of these buttons: . Draw the object by dragging the mouse cursor.
- Newly created drawing objects are added to the selected diagram. When a diagram is moved to a new position, the drawing objects are moved as well. Drawing objects of different diagrams cannot be grouped. Drawing objects can be cut and pasted to another diagram without changing their position.
- Diagrams, drawing objects and datasets can be copied using the clipboard. Use the buttons to copy and paste an object between different UniPlot diagrams or documents.
- To insert a UniPlot document page into another application, choose *editcopy-page*. This will copy all elements of the page as a Windows Metafile into the clipboard. Switch to the application (`Alt+Tab`) into which you want to insert the UniPlot graphic and then choose *editpaste*.
- To select several drawing objects at the same time, hold down the `Shift` key while you click an object. To cancel a selection, click the object while you hold down the `Shift` key.

- To group selected drawing objects to a single unit, click the button. With the button you can ungroup a drawing object. Only drawing objects belonging to one diagram can be grouped.
- Selected drawing objects, diagrams and datasets can be deleted with the **Delete** key. If you delete a diagram, all of the datasets and the attached drawing objects will be removed from the document page.
- To magnify part of a document, click the button and drag to create a rectangle that encloses the part of the page you want to enlarge.
- You can change isoline distribution and the position of isoline labels with the 10, , , and buttons. To use one of these functions, a 3D dataset must be selected. If only one 3D dataset is available, it will be automatically selected. If a diagram has more than one 3D dataset, you have to select the dataset you want to edit. To select a 3D dataset, click one of its isolines.
- To label isolines, choose the 10 button. Drag the cursor to create a line which crosses at least one isoline. Labels will be inserted at the intersection points of the line and the isolines. To insert a single label, click a point on the isoline where you want the label to appear.
- To remove isoline labels, click the button. Drag a rectangle that encloses all of the unwanted labels. To remove a single label, click it.
- To insert a new isoline into the map, select the button. Position the cursor where you want to insert an isoline and click once. A dialog box will appear in which you can edit the z-value of the isoline. To insert the line, choose the *OK* button.
- To delete an isoline, choose the button and click the isoline you want to remove.
- With the 2D and 3D buttons you can switch between a 2D and a 3D diagram. Only 3D datasets will be plotted as a surface map in the 3D diagram.
- To change the 3D view angle or perspective, choose the button.
- Diagramm 1 This combo box shows the selected diagram name. To choose another diagram, open the combo box and click another name.
- The button opens a dialog box with the names of the datasets attached to the selected diagram.
- The import function in UniPlot works somewhat unconventionally: A Text file (ASCII) or Excel file with data arranged in columns will be converted into a UniPlot data file. In a second step, the data can be easily and quickly imported from the UniPlot data file (`.nc` or `.nc2`). The import function is written in UniScript and can be adjusted to suit your needs.
- To import data from a Text or Excel file, choose *file-import-data*. Depending on the file size, the conversion to the UniPlot data file can take anywhere from a few seconds to several minutes.
- c:/uniplot/samples/map2.nc This box shows the names of the last 20 imported data files. To open a data file, open the list and select a file name. In the dialog box that appears, you can choose the data you want to import into the selected diagram:



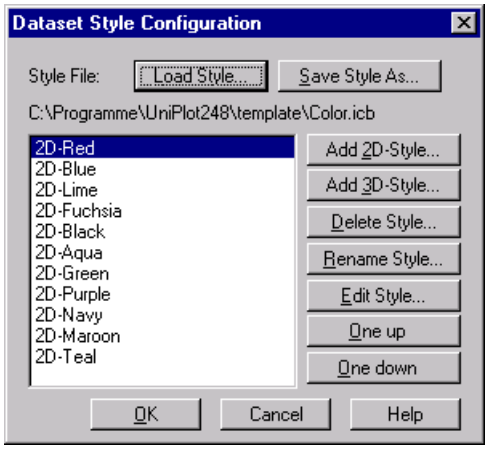
To delete the list contents, choose *filemore-file-functions* and select **Reset recent NC-data name list**.

- The data cursor for 1D and 2D datasets can be switched on with the button. To switch off the cursor, click the button again. Use the mouse to move the cursor lines. The measurement at the cursor position will be displayed in a window:



Name	Cursor 1	Cursor 2	C2 - C1	x-PosMax	y-Max	Mean	Peak to Peak
x-Position	22.2500...	22.500...	0.2500...				
b06026.nc (Col2, M_ERAUCH):SubSet	1551.690...	1787.60...	235.910...	22.50	1787.60	1681.28	235.91
b06026.nc (Col2, M_LTL):SubSet	552.000...	570.000...	18.000...	22.43	570.00	561.46	18.00
b06026.nc (Col2, M_ESTADREH):SubSet	45.0000...	52.250...	7.2500...	22.48	52.25	48.74	7.25
b06026.nc (Col2, M_ERAUCH):SubSet	36.0000...	37.500...	1.5000...	22.43	37.50	36.84	1.50
b06026.nc (Col2, M_E):SubSet	34.7500...	36.750...	2.0000...	22.46	36.75	35.95	2.00

- The button changes the axes scaling. Select a diagram. Click the button and drag a rectangle around the area of interest. The diagram axes will be scaled to show the selected area.
- To auto scale the axes, click the button. The axes will be scaled so that all datasets are completly visible inside the diagram. You can use this function to undo any zoom operation.
- With the , and buttons, periodical signals or time signals can be scrolled cycle by cycle. Import a signal and zoom a part of the signal ( ). Click on the button and specify the cycle length (e.g. 720 grad CA). The signal can be scrolled one cycle to the left or right with the and buttons.
- **Undo**: To undo any changes, choose the button. Up to 255 steps can be undone.
- **Redo**: To redo any undo steps, click the button.
- 2D-Red The **Dataset Style** box can be used to set the style of all selected datasets. To change the style of a dataset, select the dataset, open the combo list box and choose a style. To edit the styles in the list box, choose *toolsdataset-style-configuration*.

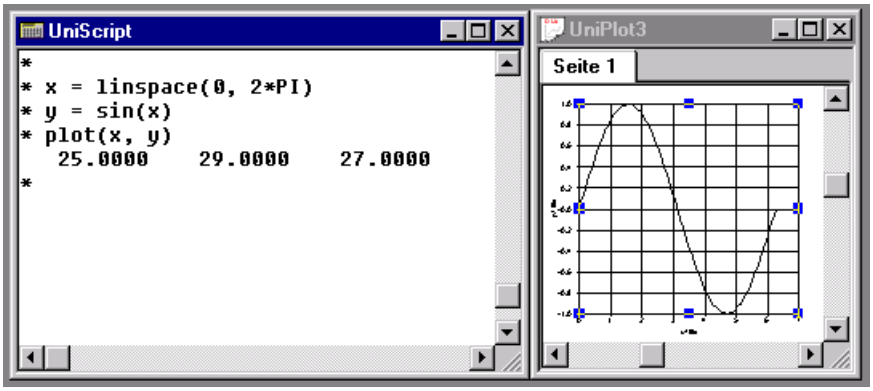


- The button opens the UniScript command window.

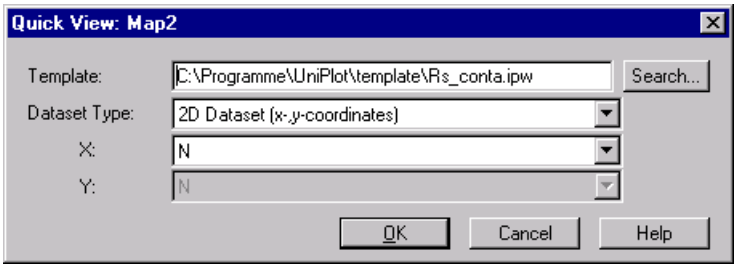
  Type in the following example:

```
x = linspace(0, 2*PI)
y = sin(x)
plot(x, y)
```

  plot is a function written in UniScript which creates a new document with a diagram and plots data y versus x into a diagram. You can look into the source code of the *plot* function. Open ( ) the `plot.ic` file in the `UniPlot\Script` directory. To get help for one of the functions, position the cursor on a function name (e.g. *XYCreate*) and press **F1**.



- The **File Quick View** function can be used to get an overview of a data file. Choose *filefile-quick-view*. In the dialog box select a data file. The file will be converted and the following dialog box will be displayed:



  Any UniPlot file (`.IPW`) can be used as a template for the quick view function. The default template

is the file `UniPlot\Template\rs_conta.ipw`.

CHAPTER 4

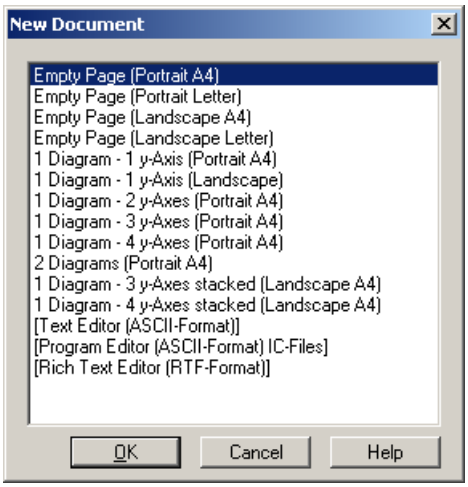
A Simple Example

The following example will help familiarize you with some of UniPlot's functions.

Task: To produce an isoline diagram and a 2D diagram with a Full Load Line (Wide Open Throttle curve or WOT).

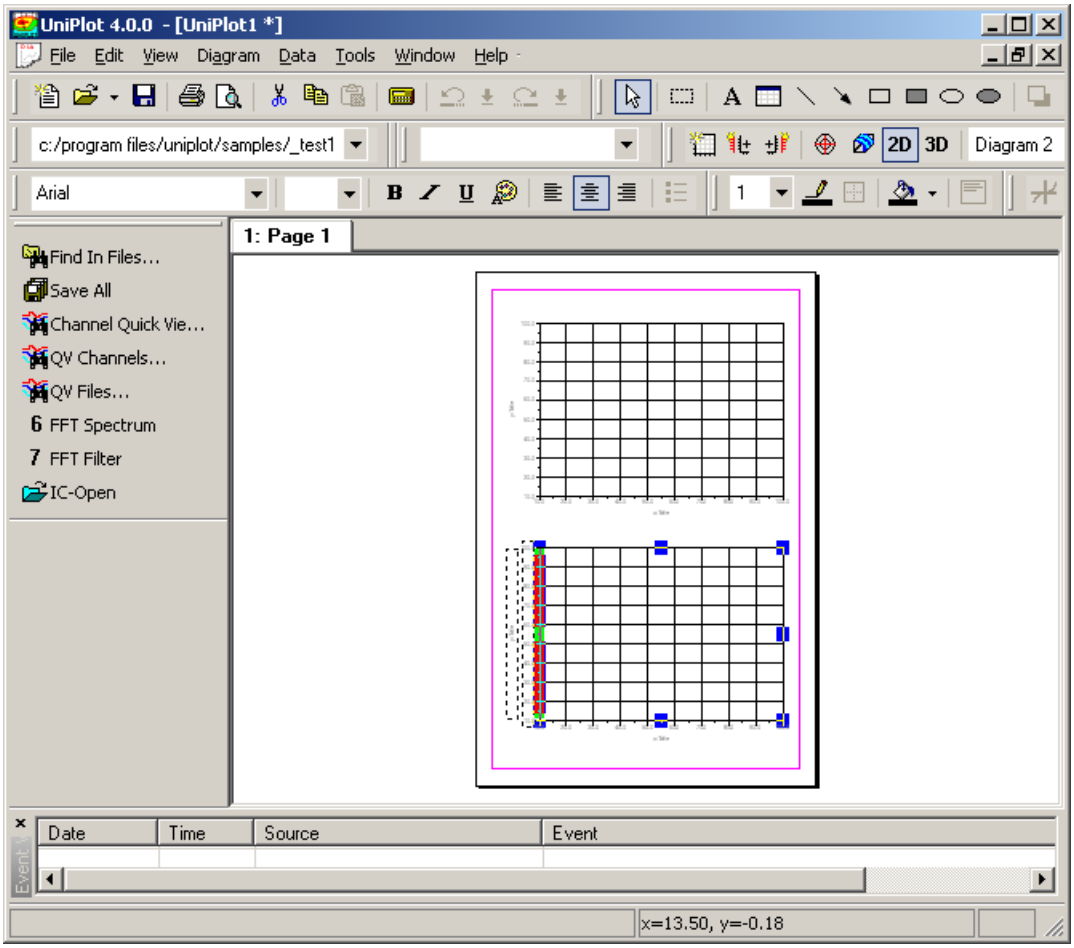
4.1 Creating a Document

In the first step, we will create a new document with two diagrams. Click the **New** button in the toolbar or choose *filenew*.


New Document
Empty Page (Portrait A4)
Empty Page (Portrait Letter)
Empty Page (Landscape A4)
Empty Page (Landscape Letter)
1 Diagram - 1 y-Axis (Portrait A4)
1 Diagram - 1 y-Axis (Landscape)
1 Diagram - 2 y-Axes (Portrait A4)
1 Diagram - 3 y-Axes (Portrait A4)
1 Diagram - 4 y-Axes (Portrait A4)
2 Diagrams (Portrait A4)
1 Diagram - 3 y-Axes stacked (Landscape A4)
1 Diagram - 4 y-Axes stacked (Landscape A4)
[Text Editor (ASCII-Format)]
[Program Editor (ASCII-Format) IC-Files]
[Rich Text Editor (RTF-Format)]
OK
Cancel
Help

In the **New Document** dialog box, select the **2 Diagrams (Portrait)** option and click the OK button or just double-click the item.

A new window with a portrait oriented page and two diagrams will be created. The page and pink margins will help you to position the diagrams and the drawing objects. These elements will not be plotted on the printer. To change page margins, choose *filepage-margin*. To change the paper size and orientation, choose *filepage-setup*. The options in the Page Setup dialog box depend on the capabilities of the selected printer.



You can change the Page Setup at any time. However, the position of drawing objects and diagrams will not be adjusted automatically to the new settings.

4.2 Importing Data into a Diagram

In this step we will import a 3D dataset.

A 3D dataset can either be created from x, y, z data that is irregularly distributed in the x/y plane or from data that is already in a matrix form. In this example, we will create the dataset from the fuel consumption data of an engine versus speed and torque with an irregular distribution.

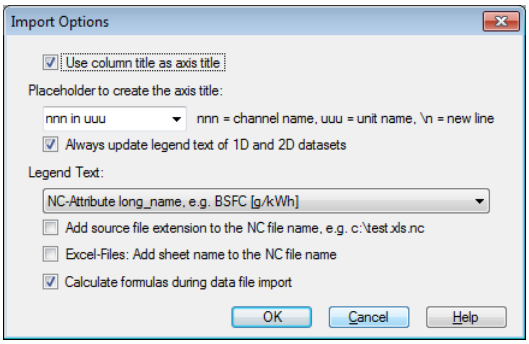
Import Options: Before we import the data, we will choose *fileimport-options*. In this dialog box, you can select column separators for Text files. The columns can be separated by blank and tab characters, one comma, one semicolon or one tab character. For text files, the decimal separator can appear as a point (`"."`) or a comma (`","`).

If the data columns have names above the data points, the name can be loaded and used to label the axis and the datasets.

Example data files are located in the `uniplot\samples` directory. To see how the data files are formatted, load one data file into an editor. Choose *fileopen* and load the `map2.asc` file. The data columns in this file are separated by blanks and the decimal separator is a point. The names of the columns appear 2 rows

before the first data row. Before loading the data into a diagram you should close the editor with the *fileclose* function.

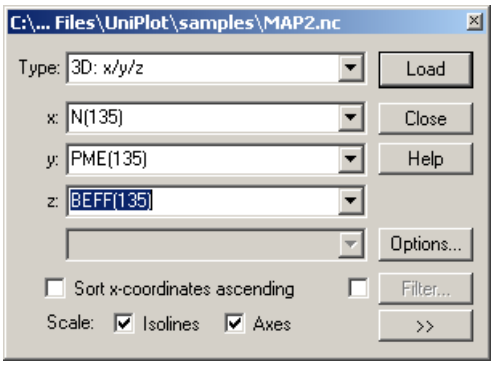
For the `map2.asc` data file, information should be entered into the *fileimport-options* box as shown below.


Import Options
Use column title as axis title
Placeholder to create the axis title:
nnn in uuu
nnn = channel name, uuu = unit name, \n = new line
Always update legend text of 1D and 2D datasets
Legend Text:
NC-Attribute long_name, e.g. BSFC [g/kWh]
Add source file extension to the NC file name, e.g. c:\test.xls.nc
Excel-Files: Add sheet name to the NC file name
Calculate formulas during data file import
OK
Cancel
Help

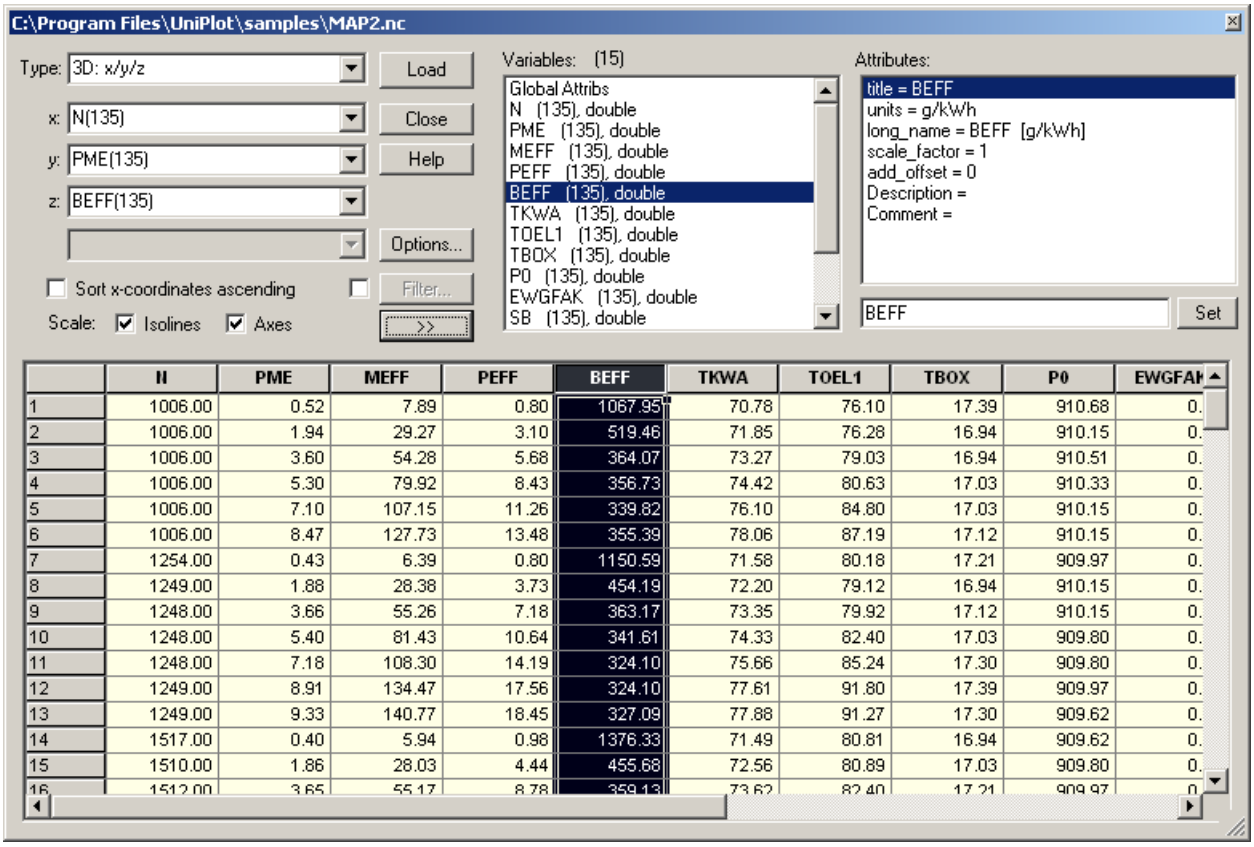
After you have set the options, choose *fileimport-data*. In the dialog box, select the `map2.asc` file from the `uniplot\samples` directory. The file will be opened and converted into a netCDF file. The name of the netCDF file will remain the same but the extension will be changed to `.nc`.

If a file with the same name already exists, a message box appears. The existing file can be written over or the new file can be saved under a different name.

If the text file was successfully converted to a netCDF file, the Import dialog box will appear.



The list of variables **X**, **Y**, **Z** shows the converted data columns. Only the columns with values will be converted. Any other columns, such as time or date columns, will be skipped. The only exception is the definition of the Full Load Line (WOT) for 3D datasets.



	N	PME	MEFF	PEFF	BEFF	TKWA	TOEL1	TBOX	P0	EWGFAK
1	1006.00	0.52	7.89	0.80	1067.95	70.78	76.10	17.39	910.68	0.
2	1006.00	1.94	29.27	3.10	519.46	71.85	76.28	16.94	910.15	0.
3	1006.00	3.60	54.28	5.68	364.07	73.27	79.03	16.94	910.51	0.
4	1006.00	5.30	79.92	8.43	356.73	74.42	80.63	17.03	910.33	0.
5	1006.00	7.10	107.15	11.26	339.82	76.10	84.80	17.03	910.15	0.
6	1006.00	8.47	127.73	13.48	355.39	78.06	87.19	17.12	910.15	0.
7	1254.00	0.43	6.39	0.80	1150.59	71.58	80.18	17.21	909.97	0.
8	1249.00	1.88	28.38	3.73	454.19	72.20	79.12	16.94	910.15	0.
9	1248.00	3.66	55.26	7.18	363.17	73.35	79.92	17.12	910.15	0.
10	1248.00	5.40	81.43	10.64	341.61	74.33	82.40	17.03	909.80	0.
11	1248.00	7.18	108.30	14.19	324.10	75.66	85.24	17.30	909.80	0.
12	1249.00	8.91	134.47	17.56	324.10	77.61	91.80	17.39	909.97	0.
13	1249.00	9.33	140.77	18.45	327.09	77.88	91.27	17.30	909.62	0.
14	1517.00	0.40	5.94	0.98	1376.33	71.49	80.81	16.94	909.62	0.
15	1510.00	1.86	28.03	4.44	455.68	72.56	80.89	17.03	909.80	0.
16	1512.00	3.65	55.17	8.78	359.13	73.62	82.40	17.21	909.97	0.

To open it to its full size click the button. The key word "`double`" in front of the variables means that the data points of the variable are float numbers with double precision (8 Bytes). The number in brackets behind the variable is the number of data points.

The data points of the selected variable are listed in the left list box of the dialog box. To edit a value in the list box, select the data point in order to copy the value into the text field. To set the data point to a new value, choose the **Set** button.

The attributes of each variable are displayed in the attributes list box underneath the variables list box.

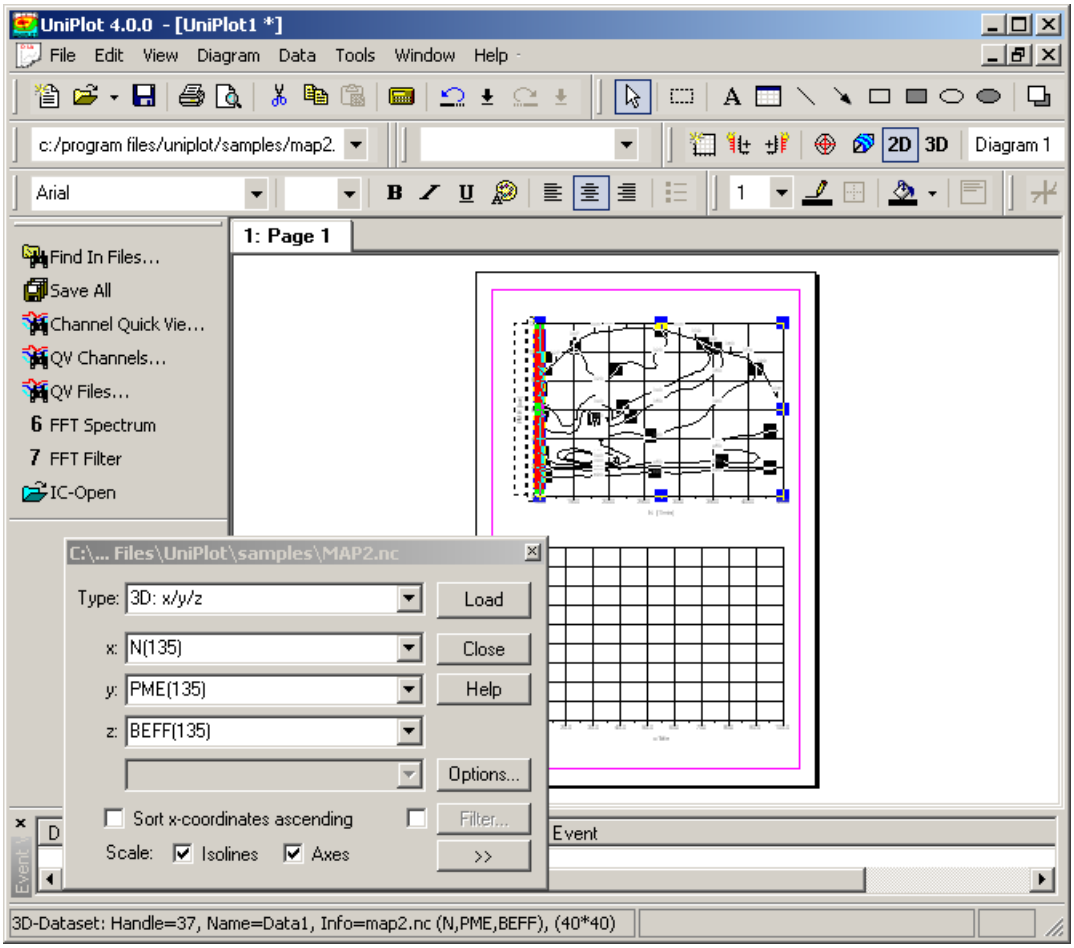
The variable **Global Attribs** hasn't any data points. This variable has only attributes. The Range attribute has a special meaning for 1D and 2D datasets. It consists of two values separated by a comma. The first value is the index of the data point where the import should start and the second value is the number of data points that will be imported. Default is "`1, Total Number of Points`".

Caution: Any changes to data points and attributes will be saved immediately in the netCDF data file.

In the Reading Instructions dialog group, you can select the type of the dataset and the data columns you want to import. To import a 3D dataset, click the 3D radio button. The isoline and axes check boxes should be selected in order to scale the axes and isolines automatically.

Before importing a dataset, we need to select the diagram the dataset should be assigned to. Click inside of the top diagram. The name of the selected diagram will be displayed in the toolbar.

Now we are ready to import the 3D dataset to create a fuel consumption map. Assign the variable `N` to the x-axis, the variable `PME` to the y-axis and the variable `BEFF` to the z-axis. When you have set the variables, select the **Load** button to import the data into the selected diagram.



During the import, data points will be read from the file and saved in a dataset object. An equidistant matrix with z-values will be computed from the imported data points. In the next step, isolines will be calculated from the interpolated matrix. As well as the original data and the interpolated matrix, other information will be stored in the dataset object i.e. the isoline values, fill colors, label fonts, the dataset hull, label format, etc. This information will be initialized with default values during the creation of the dataset. Default values for dataset objects, diagrams and drawing objects can be set in the **Tools** menu.

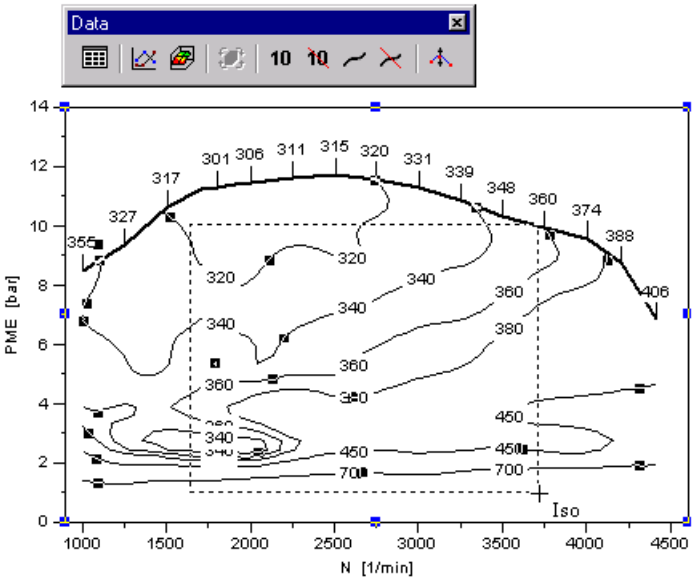
Isolines and their labels will be created automatically. In the next step, we will change their distribution and the position of the labels.

4.3 Editing Isolines

Parts of the document can be magnified using the **Zoom** function to enable easy editing of isolines. Choose the **Zoom In** button and drag a rectangle around the part of the diagram to be enlarged. The **Zoom Out** button will reduce the document size to fit into the window. If you click the **Zoom Out** button repeatedly, you can switch between the full size window, 50% of the window size and the original size.

In the first step, we will delete the isoline labels. Choose the **Delete Label** button with the crossed out 10. The mouse cursor will change to a cross with the label **Iso**. Position the cross cursor in the upper left corner of the diagram and drag a rectangle that encloses all of the isoline labels. The labels inside the

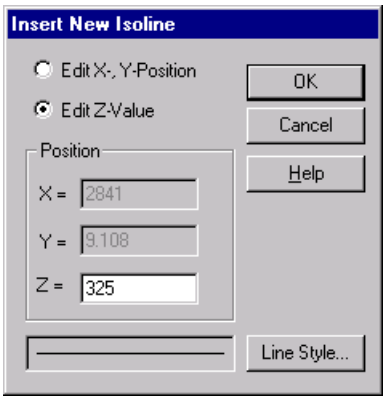
rectangle will be removed from the map.



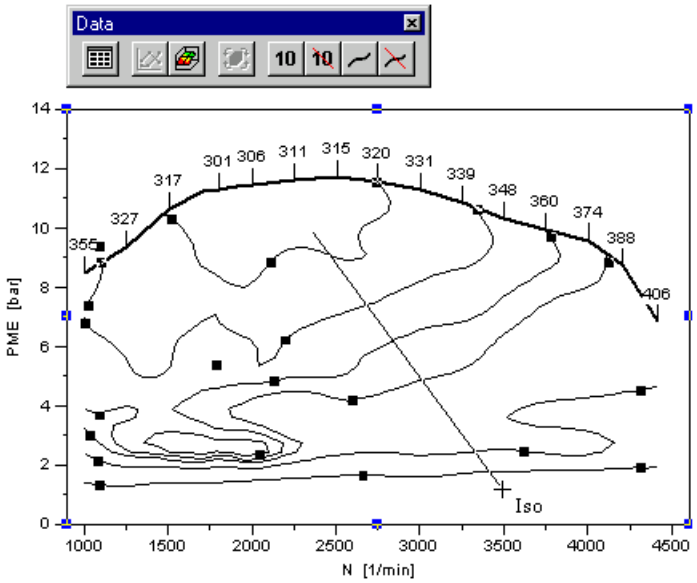
Any mistakes that might have been made inserting and deleting isolines and their labels can be eliminated using the **Replot** button from the toolbar or press `Ctrl+R`.

Next, we will add new isolines to the map. Choose the button from the toolbar. Click a point on the map with the cursor. You can only insert lines inside the data hull of the map. If you click outside of the map, UniPlot will switch back to the Selection mode and the cross cursor will return to the shape of an arrow cursor .

If you click inside of the map, a dialog box with the x, y, z coordinates of that point will appear. Choose *OK* to insert the line into the map.



To label the isolines, choose the **Insert Label** button from the toolbar. Draw a line by holding down the left mouse button and dragging the mouse. The line must cross at least one isoline. A label will appear at each point where the line intersects an isoline. To insert a single label, click a point on the isoline where you want the label to appear.



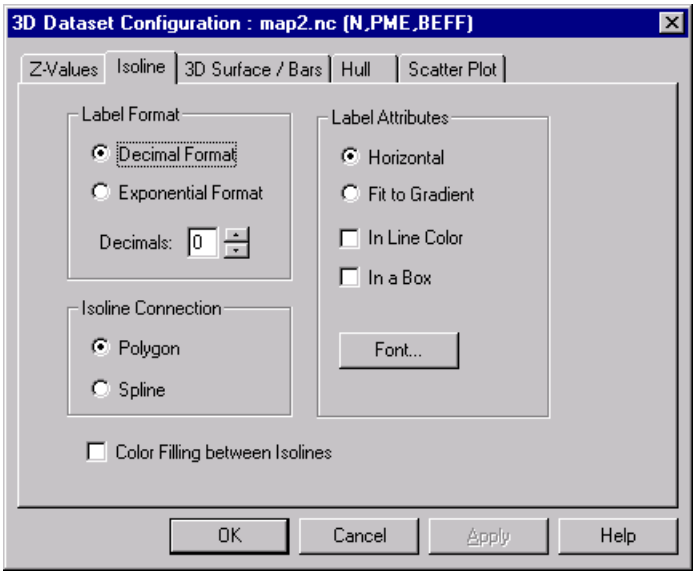
The labels on the Full Load Line (WOT) cannot be edited in the same way. Full load line label editing is covered in the chapter “Definition of the Full Load Line” .

To delete an isoline, choose the button with the crossed out isoline and click the isoline which you want to remove. An isoline can consist of more than one line segment of the same value. This function deletes all single line segments of the same value and their labels.

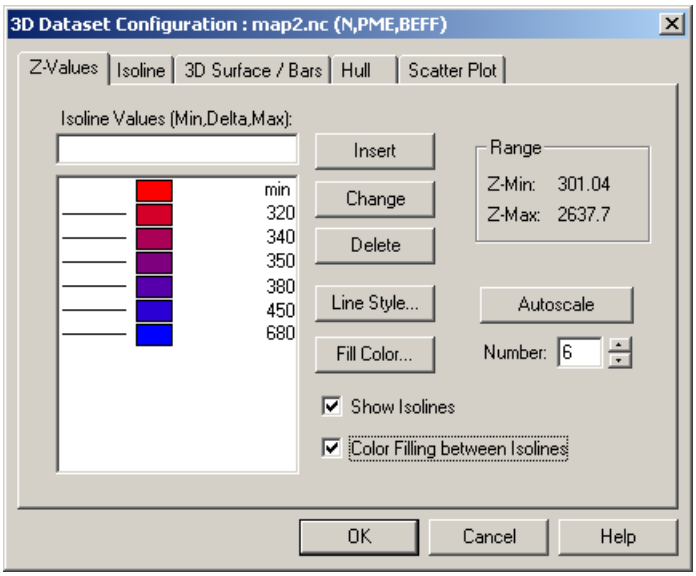
Click outside of the map to switch back to the Selection mode. The cursor will change back to an arrow.

4.4 3D Dataset Configuration

Next we will define a color gradient for the map. Double-click one isoline to open the **3D Dataset Configuration** dialog box and the select the Isoline tab. Select the **Color Filling between the Isolines** check box. In this dialog page you can also specify the format and alignment of the isoline labels.

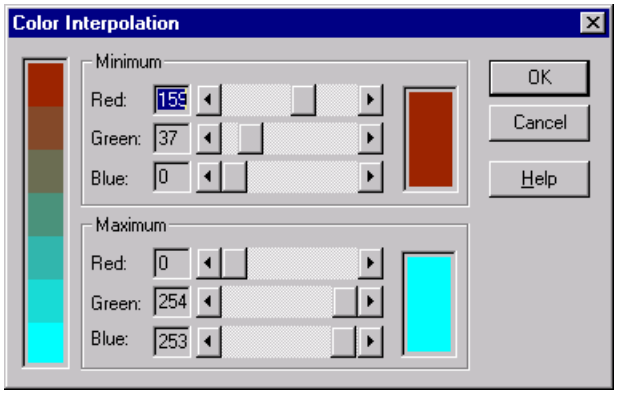


To modify the default color gradient, select the **Z-Values** tab. In the list box, the iso values of the map, the fill colors and the line style for the isolines are displayed. The values are sorted in increasing order. Fill color is used to fill the area between the value that is displayed beside the fill color and the next bigger value. To be able to define a fill color for even the smallest value, the value **min** has been inserted at the beginning of the list.

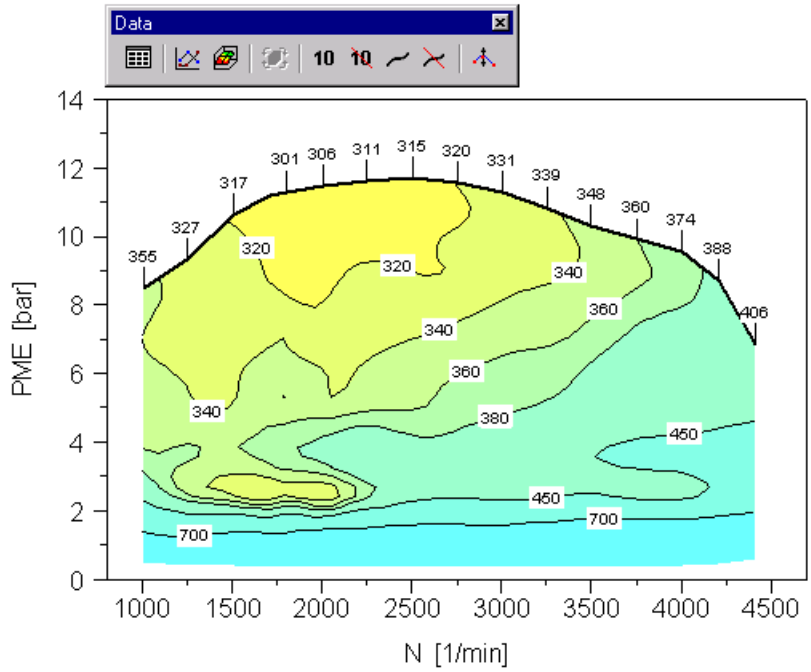


When you select a value from the list, it will be transferred to the edit field above the list. Edit the value and then click the **Change** button. The new value will be added to the list and replace the old one.

If you need help with the dialog box, press **F1** for a description of the options.

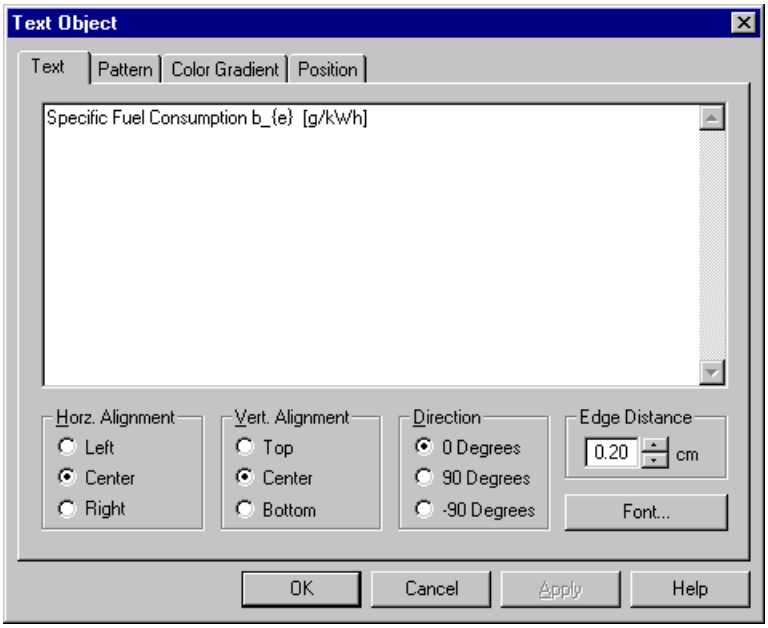


To specify a color gradient, mark all the iso values including **min** in the list and choose the **Fill Color** button to open the color gradient dialog box. You can adjust the colors for the smallest (Minimum) and the largest (Maximum) value by dragging the scroll box. The color gradient will be displayed in the left field of the dialog box. Choose *OK* to apply the new color gradient to the iso values in the list box.



4.5 Diagram and Axes Titles

To create a title for the document, choose the **Text** button [A] from the toolbar. The cursor will change to a cross with the label **new**. Drag a rectangle above the upper diagram to specify the size of the text box. Double-click the new text object to open the **Text Object** dialog box.



Type the following text into the edit field:

```
Specific Fuel Consumption b_{e}  [g/kWh]
```

The text object's font and frame can be changed by selecting the appropriate button or tab in the dialog box.

If you move the cursor over the different elements of a diagram, the cursor changes to a cross with a label. For example, over the axis title, the cursor will display the text **Title**, over the axis labels, **Label** and over the diagram axis, **Axis**. When you double-click one of these elements you can open the dialog box to change the options.

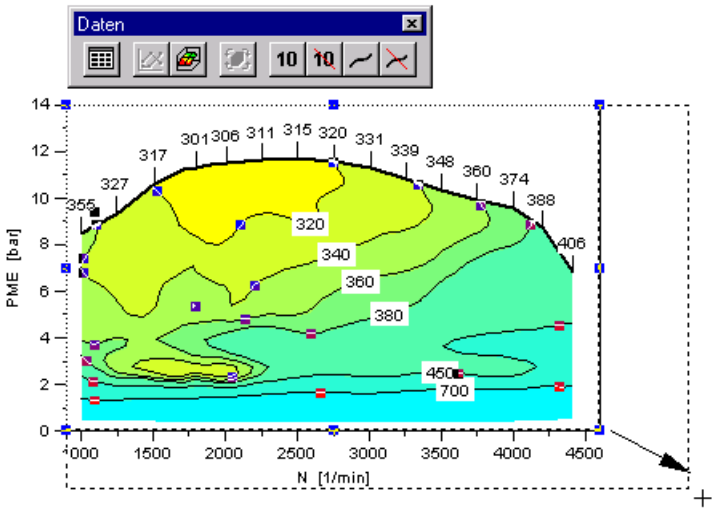
Position the cursor on the x-axis title and open the title dialog box. Type in the following text:

```
Speed [rpm]
```

and then choose the *OK* button.

4.6 Editing Diagram Configuration

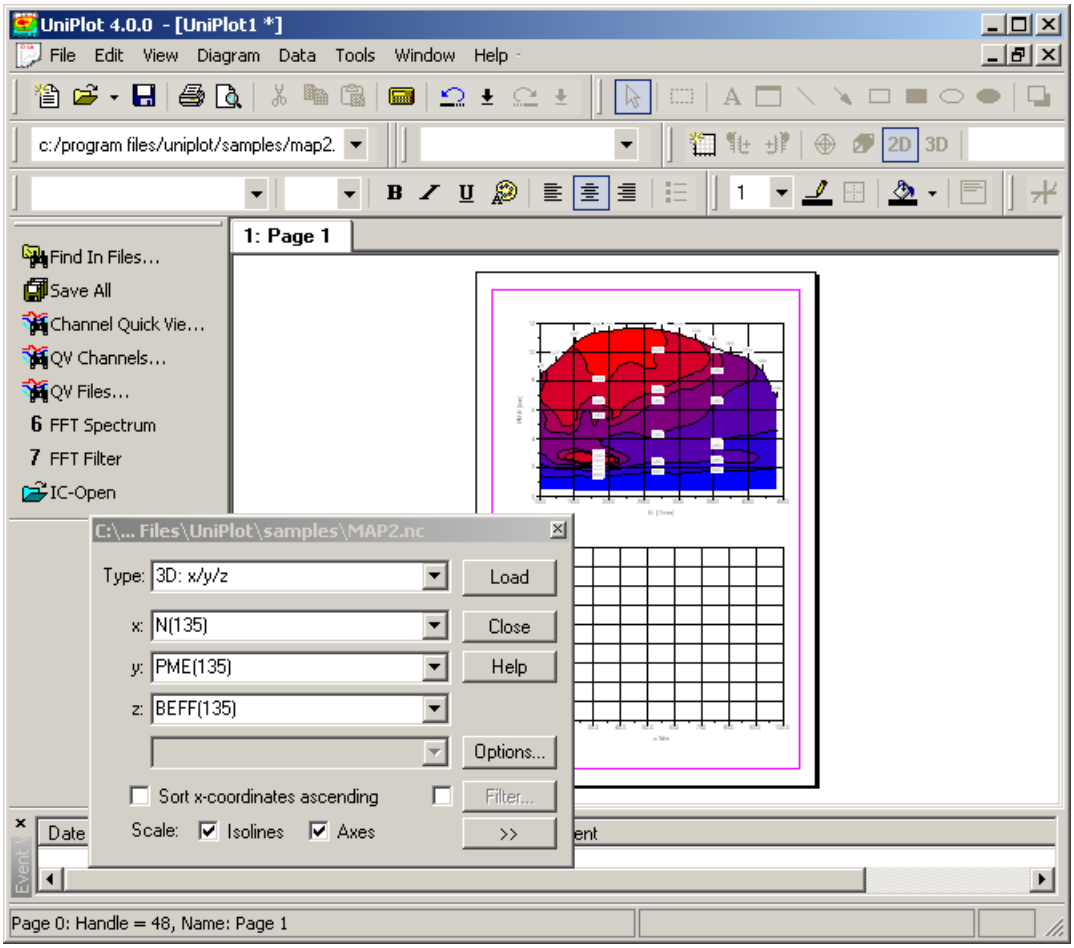
To resize a diagram, drag one of the blue handles. To move a diagram to a new position, grab the diagram between the blue handles. When the cursor is over the frame of the diagram, it will change to a cross. The size of the diagram will be displayed in the status bar in centimeters.



4.7 Importing a 2D Dataset

To import a dataset into the lower diagram, click inside the frame of the diagram with the name **Diagram 2**. The name of the diagram will be displayed in the combo box in the toolbar. To load a 2D dataset, choose *fileimport-data*. In the **Open** dialog box, choose the `vollast.asc` file and then choose the *OK* button.

The file will be converted to a netCDF file and the **Import** dialog box will be displayed. Choose the **2D** button and select `N` (speed) for the x-column. For the y-column, choose `PME`. To create the dataset, choose the **Load** button.


UniPlot 4.0.0 - [UniPlot1 *]
File Edit View Diagram Data Tools Window Help
c:/program files/uniplot/samples/map2.
2D 3D
1: Page 1
Find In Files...
Save All
Channel Quick Vie...
QV Channels...
QV Files...
6 FFT Spectrum
7 FFT Filter
IC-Open
C:\... Files\UniPlot\samples\MAP2.nc
Type: 3D: x/y/z
Load
x: N(135)
Close
y: PME(135)
Help
z: BEFF(135)
Options...
Sort x-coordinates ascending
Filter...
Scale: Isolines Axes
>>
Date
Page 0: Handle = 48, Name: Page 1

CHAPTER 5

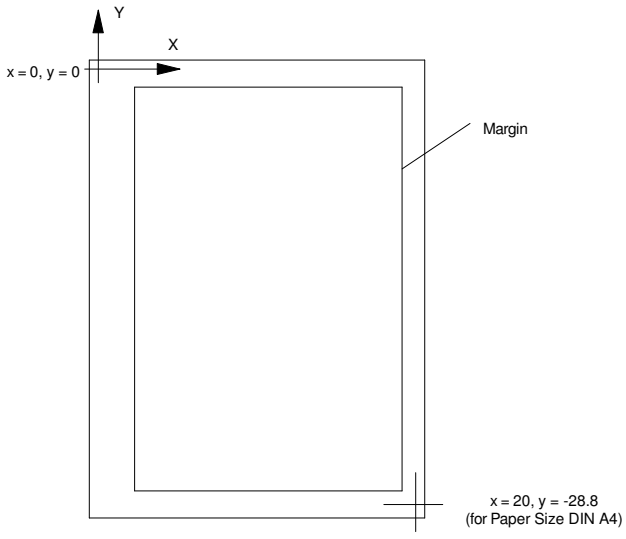
UniPlot Objects

5.1 UniPlot Documents

A UniPlot document can consist of up to 255 pages. The active page will be displayed on the monitor in the right size ratio. The pink rectangle inside the paper frame is the page margin which can be set in centimeters in the **Page Margin** dialog box *filepage-margin*.

For each page a tab register with the page name is displayed. To add a new page to the document, choose *editpagenew-page*.

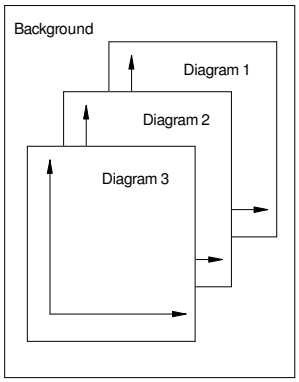
The following figure shows the origin of the page coordinate system:



Paper Size and Paper Orientation can be set in the Page Setup dialog box *filepage-setup*. The paper sizes that are available depend on the capabilities of the printer you have selected. The page can be oriented vertically (portrait) and horizontally (landscape). Page settings will be saved in the document.

The figure above shows the page in portrait orientation with a page margin and a coordinate system which is used to position the objects on the page. The x-axis of the coordinate system points to the right, the y-axis points up. The origin is always in the upper left corner of the page. The coordinates are measured in centimeters.

A page in UniPlot is built from diagrams which are placed on the page. Each diagram can hold datasets and drawing objects. The drawing objects are placed in front of the diagram and the datasets. When you add a new drawing object to a diagram, it is placed in front of all other attached drawing objects. You can change the layering order of diagrams with the **One Layer up**, **One Layer Down**, **Bring To Front** and **Send To Back** buttons. You can also change the layering order of drawing objects and datasets with these buttons.



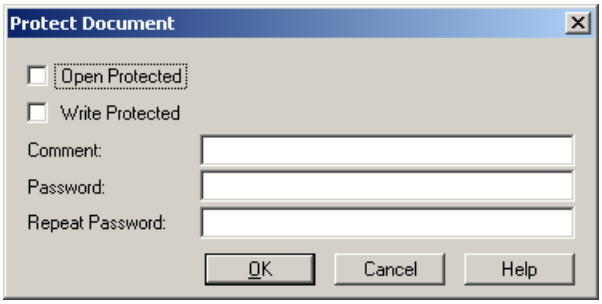
To place a drawing object behind the diagrams, it must be added to the background layer.

UniPlot draws the objects in the following order (from back to front):

- Drawing objects in the background
- Diagram rectangle
- Datasets (Fill color, hatch)
- Diagram grid
- Axes with labels and titles
- Datasets (line and marker)
- Drawing objects of the diagram

5.1.1 Document Protection

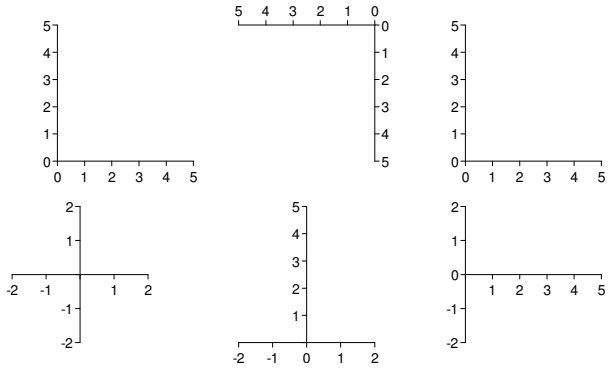
One way of controlling access to information in UniPlot documents is to set passwords. The *toolsprotect-document* dialog box provide the option to assign a password. When you set either open or modify protection, the password and the contents of the document are not encrypted. If you need higher level of security you should use a different method to protect your documents.



5.2 2D Diagram

In UniPlot, a 2D diagram has one x- and one y-axis. The axes are perpendicular to each other. x-values increase to the right and y values increase upwards. If necessary, the axis can be scaled in decreasing order. The y-axis can be placed on the left or right side of the diagram. The x-axis can be placed at the top or the bottom of the diagram. The axes can be hidden if you do not want them to be drawn. They can also be moved parallel to their standard positions.

The following figure shows some examples:



From a single diagram more complex diagrams with several axes can be created. The following picture shows some examples of diagrams with multiple axes.

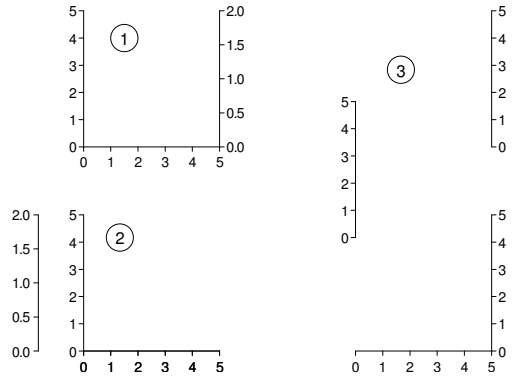


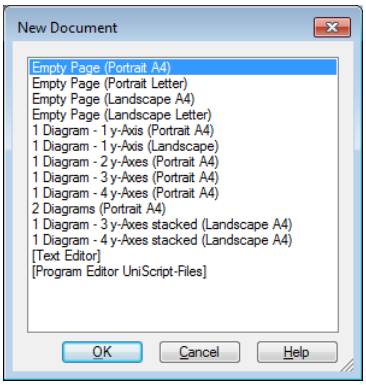
Figure 1 has two y-axes and one x-axis. It is created from two overlapping diagrams. One diagram has the y-axis on the left side and the x-axis at the bottom. The other diagram has its y-axis on the right side and the x-axis is hidden.

Figure 2: y-axes are stacked on the left side of the diagram.

Figure 3 is created from 3 single overlapping diagrams. Only the lower diagram has an x-axis. The x-axis is hidden in the two upper diagrams and linked to the bottom diagram.

Axes of the same type can be linked, therefore they will use the same axis scaling. Linked axes are especially useful for diagrams with multiple y-axes. In **figure 3**, the x-axes of the upper two diagrams are linked to the x-axis of the bottom diagram. When the scaling of one axis is changed, the linked axes will be rescaled automatically. Only the scaling of an axis is affected by the link. The diagram title, the axis length and other settings of the diagram are not affected through an axis link. To link axes of different diagrams in one page, choose the **Diagram Setting** dialog box (*diagramdiagram-settings*). A function written in UniScript

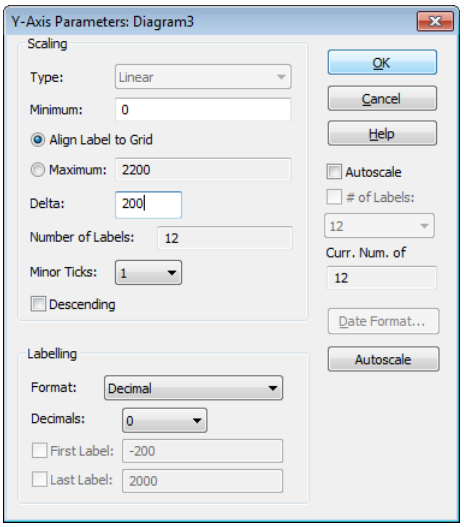
is available to create diagrams with multiple y-axes. To execute this function, create an empty page and choose *diagrammore-diagram-functions* to open the following dialog box:


New Document
Empty Page (Portrait A4)
Empty Page (Portrait Letter)
Empty Page (Landscape A4)
Empty Page (Landscape Letter)
1 Diagram - 1 y-Axis (Portrait A4)
1 Diagram - 1 y-Axis (Landscape)
1 Diagram - 2 y-Axes (Portrait A4)
1 Diagram - 3 y-Axes (Portrait A4)
1 Diagram - 4 y-Axes (Portrait A4)
2 Diagrams (Portrait A4)
1 Diagram - 3 y-Axes stacked (Landscape A4)
1 Diagram - 4 y-Axes stacked (Landscape A4)
[Text Editor]
[Program Editor UniScript-Files]
OK
Cancel
Help

Select **Add diagram with multiple y Axes**. This function creates a diagram with up to 8 y-axes.

Tip: To move all diagrams on the page to a new location at the same time, select *diagram-select-all-diagrams* and then move the diagrams using the arrow key or the mouse.

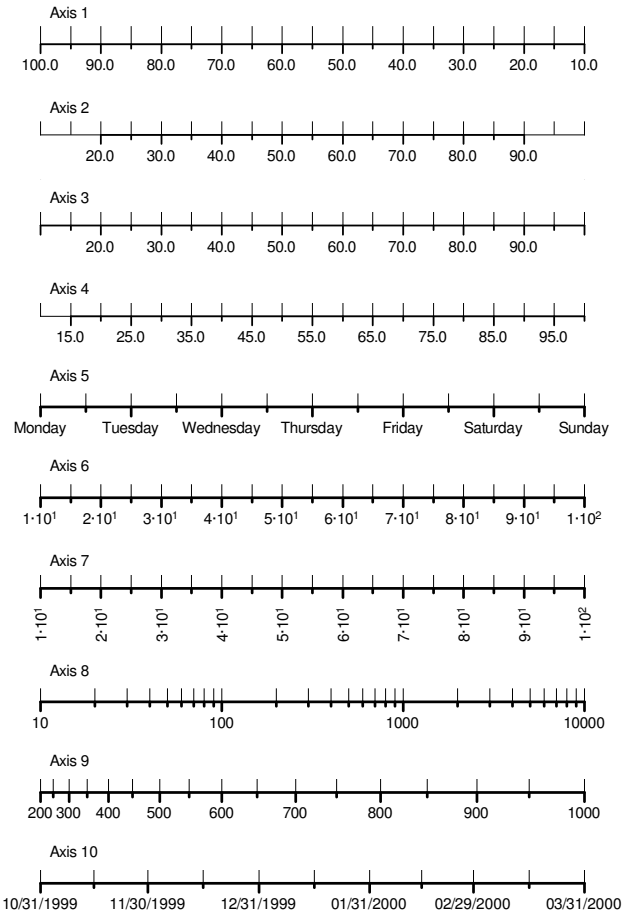
Axis scaling and label formats can be adjusted in the **Axis Parameters** dialog box. The range of the axis can be specified through the Minimum and Maximum values. The **Delta** parameter specifies the distance between the axis labels. Each label consists of a tick mark and the value of its position.



If the axis should start with a value greater than **Minimum**, click on the **First Label** check box and type its value into the text box. If the axis should end with a value smaller than **Maximum**, click on the **Last Label** check box and type its value into the text box.

Axes are normally scaled in ascending order. You can change this order to descending by selecting the **Descending in the Scaling group** check box.

The following figure shows differently scaled axes:



Axis 1 is scaled in descending order.

The minimum parameter for **Axis 2** was set to 10 and the maximum parameter was set to 100. The value of the first label was set to 20 and the last label 90.

For **Axis 3**, the boxes for **First Label** and **Last Label** were not checked. In order for the first axis label to begin at 20 even though the axis begins at 10, the text for labels 10 and 100 were deleted from the **Label** dialog box for this example.

For **Axes 5** and **7**, labels were created (10 to 70) and days of the week were chosen for the label text in the **Axis Label** dialog box.

For **Axis 6**, the exponential format was chosen in the **Axis Parameters** dialog box.

For **Axis 7**, the labels were rotated by 90 degrees in the **Axis Label** dialog box.

For **Axis 8**, a logarithmic scale from 10 to 10000 was chosen in the **Axis Parameters** dialog box.

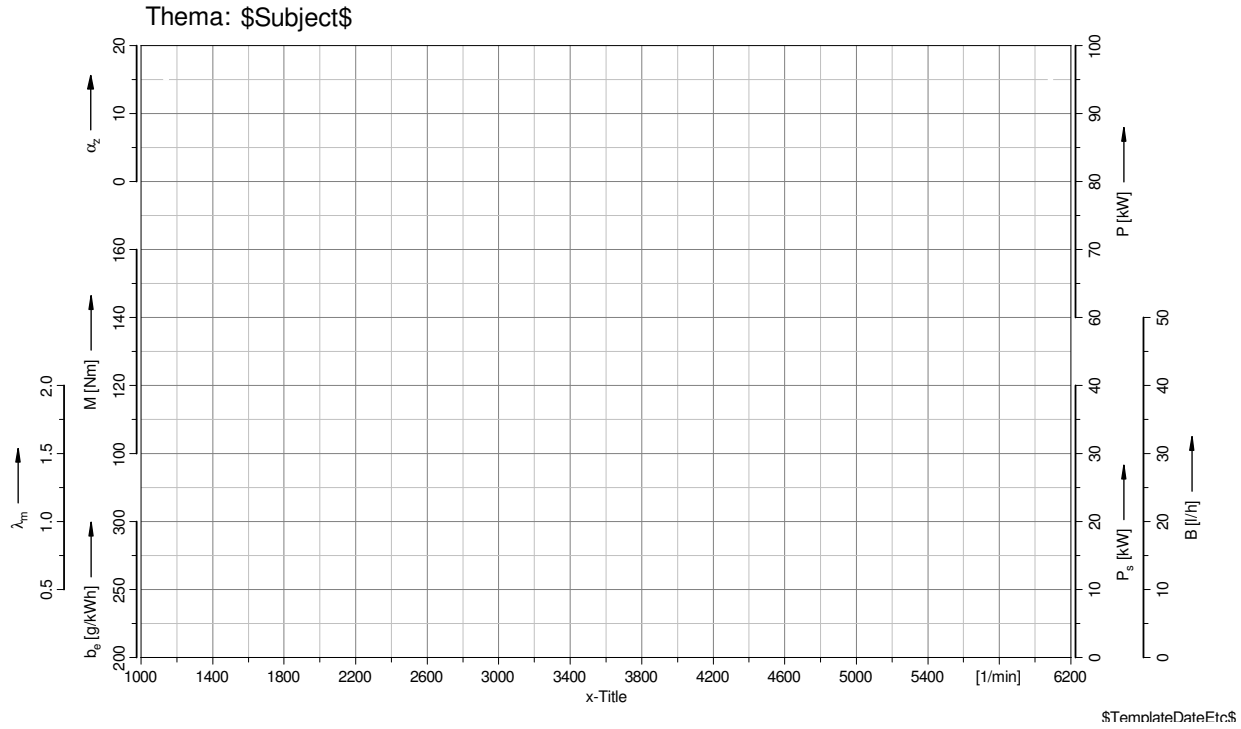
For **Axis 9**, a square scale from 200 to 10000 was selected.

For **Axis 10** a time/date scale for the 10/31/1999 to 03/31/2000 with a step of one month was selected.

Diagrams with multiple axes can be configured as stacked diagrams

5.3 Stacked diagrams with multiple y axes

The following figure shows a stacked diagram. The y-axes are arranged on both sides of the grid. The diagram has one x-axis at the bottom.



A stacked diagram is created from multiple overlayed diagrams (layers). Each layer displays a y-axis on the left or right side of the diagram. The main advantage of the stacked diagram is that the y-axis scaling will always correspond with the grid lines. Therefore the axis scaling can only be changed in steps so that the axis labels and tick marks correspond with the grid. The y-axis can only be scaled **linear**.

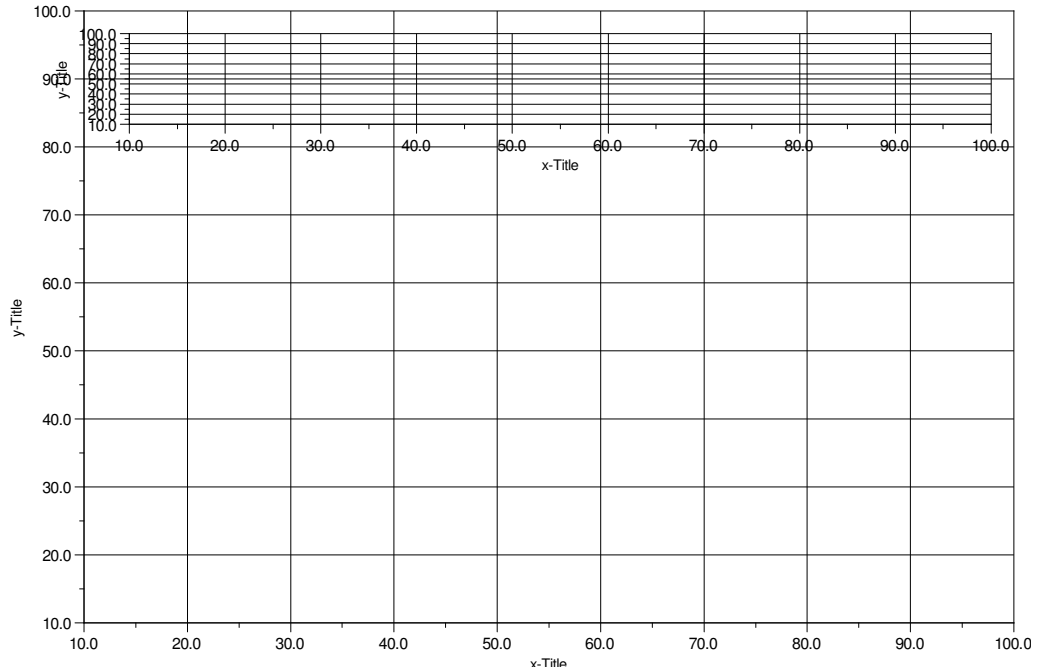
Every diagram has one layer which displays the grid lines. Normally this layer is placed at the bottom of the Z-order. The grid layer should not contain any datasets. The grid layer displays the x-axis. All other x-axes are hidden and linked to the grid x-axis. The grid layers name should be "grid".

One page can have multiple stacked diagrams. In this case the layer names are extended by a number, e.g. (Grid1, Grid2,.. etc.).

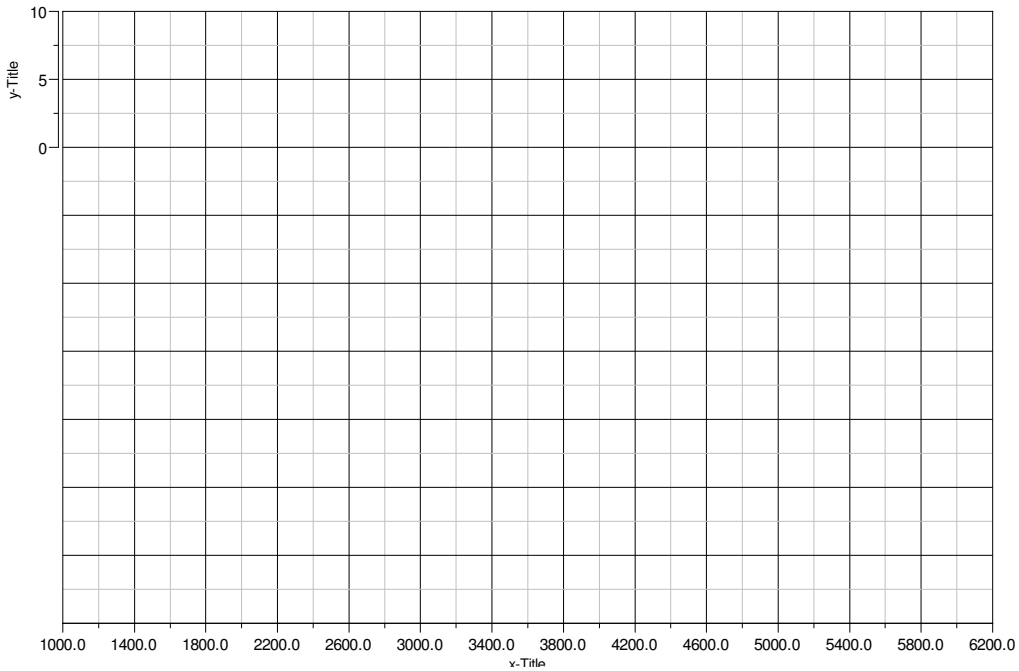
For a stacked diagram four yellow handles will be displayed in the four corners.

5.3.1 Creating a stacked diagram

- Create at least two diagrams using the button.
- Select both diagrams by holding down the `Shift` key and clicking on inside the diagram.



- Right click on the diagram and choose the command *diagram-create-stacked-diagram*.



- To shift the axis from the left to the right side, grab the axis inside the red bar and move it to the right side. If you hold down the `Shift` key will dragging the axis will be mirrored.

  Instead of eight handles (blue rectangle at the corners and in the middle of each edge) each layer has only two handles. If you grab the layer inside the handle you can change the axis length. If you grab the diagram at the edge outside the handle you can move the axis up and down.

5.3.2 Features of stacked diagrams

- All layers have the same width and share the same x-axis.
- The grid division is set in the grid layer.
- To select the grid layer click on the x-axis.
- The datasets are clipped at the diagram border if clipping is activated. To activate clipping double-click inside the layer to open the *diagramdiagram-settings* dialog box.
- One of the layers has the name "grid" or "grid1", etc.

5.3.3 Changing the diagram size

5.3.4 Modifying the grid division

To edit the grid division click on the x-axis to select the the grid layer. Click the button to open the y-axis parameter dialog box. The parameters `Minimum`, `Maximum` and `Delta` sets the axis scaling, i.e. specify the number of grid lines. If you would like a centimeter grid and the length of the y-axis is 18 centimeters set the values `Minimum = 0`, `Maximum = 18` and `Delta = 1`.

5.3.5 Adding y-axes

To add another y-axis to the diagram use the and button. To do so click on a y-axis and than click the appropriate button from the toolbar.

If you add a diagram using the button, you can add the diagram by right clicking in the diagram to open the context menu and choosing the command **Create Stacked Diagram**.

5.3.6 Shifting a y-axis

To shift a y-axis click on the axis. Move the mouse cursor into the red bar and drag the axis to its new position. To edit the axis length, move the mouse into the upper or lower blue handle and edit the axis length. The length can only be changed in the steps of the grid width.

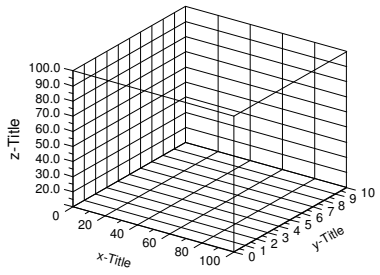
5.3.7 y-Axis Parameter

To align the axis label with the grid the option **Align Label to Grid** in the *diagramx/y/z-axisparameters* dialog box must be selected. If the option is selected the **Maximum** value cannot be set. It will be calculated from the **Minimum** value, **Delta**, axis length and the number of grid lines. Since R2012.3 this option can be disabled. The axis can still be part of a stacked diagram, but the axis range can be set freely.

Note: To move the complete stacked diagram click outside of all objects to make sure that no object is selected and then click on the x-axis to select the grid layer. Grab the diagram on the edge of the grid and move it to its new position. You can also use the arrow keys of the keyboard to move the stacked diagram.

5.4 3D Diagram

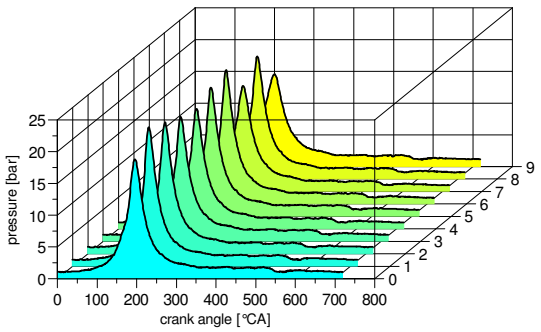
Axes positions for 3D diagrams are selected automatically. Axis elements cannot be clicked with the mouse. To change scaling, labels and the title of an axis, choose the appropriate dialog box from the *Diagram* menu or select the appropriate button from the toolbar (X , Y , Z).



Note: 3D diagrams can only be linear scaled. The axis title can only be printed as a single line.

5.5 Waterfall Diagram

In a waterfall diagram datasets are stacked into the depth of the diagram.



Data can be imported into a waterfall diagram in the same way as they are imported into a 2D diagram.

Two additional import data functions are available:

- Import of cycles from a single channel: This functions splits perodical data into cycles (*filemore-file-functionswaterfall-single-cycle-import*).
- Import of multiple channels: This function imports data from multiple channels. One channel can be defined as an x-axis channel. A dataset is created for each selected channel (*filemore-file-functionswaterfall-multi-channel-import*).

5.6 Dataset Objects

With UniPlot, 1D, 2D, 3D (triple) and 3D matrix data can be loaded:

1D Dataset: Points in a 1D dataset are measured equidistantly (e.g. Time signals or crank shaft related signals). The x-value of the first data point and the distance between two points is set in the **2D Curve Configuration** dialog box.

2D Dataset: Data points in a 2D dataset set are formed by x/y pairs. The points are added to the dataset in the same order as they are listed in the data file.

3D Dataset (triple): Data points of a 3D dataset which are formed by a x/y/z triple can be irregularly distributed in the x/y plane. To create a 3D dataset, at least 5 data points are necessary. Data points must span a surface above the x/y plane and each point in the plane can have only one z-value.

3D Matrix Dataset: The matrix must have at least 5 columns and 5 rows with z-values. For more details about different format specifications and how you can import a matrix, see **Import a Matrix** in the UniPlot Reference.

To import data into a diagram, the diagram must be selected. Click inside the diagram frame or choose the diagram's name from the toolbar.

The data import and the configuration of datasets is covered the chapter *Importing Data*.

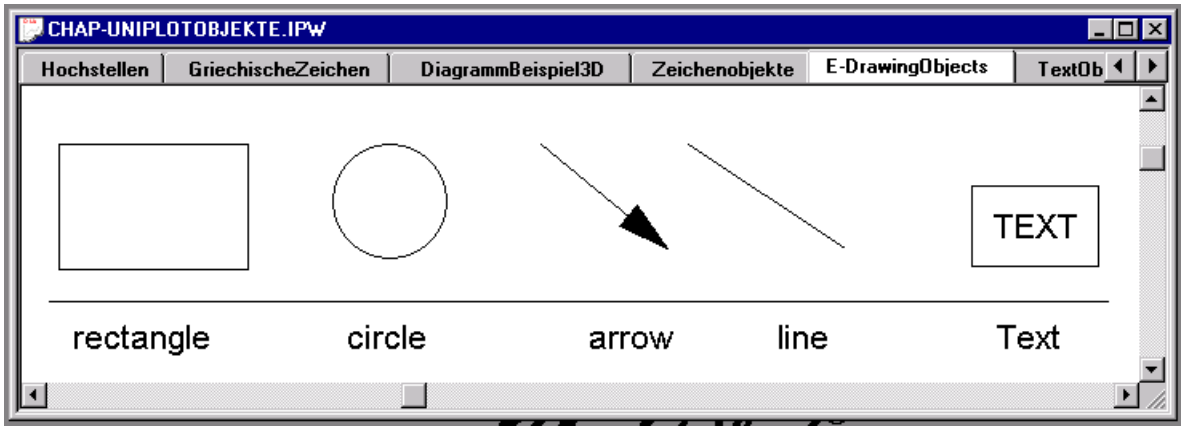
5.7 Drawing Objects

UniPlot can create lines, arrows, rectangles, ellipses, text objects and tables. To create an object, choose the appropriate button from the toolbar and drag a rectangle with the mouse where you wish to place the object. The new object will be added to the selected diagram.

5.7.1 Selecting, Resizing and Moving a Drawing Object

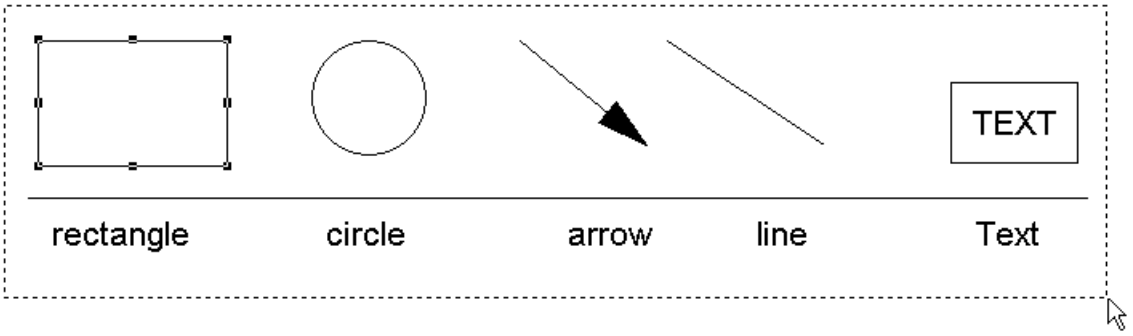
To select a single drawing object, click the object. When the cursor is on the object, the cursor will change to a cross. If the object is transparent, the object can only be selected on one of the object's lines, not within the object.

Only the drawing objects that are attached to one diagram can be selected. To select more than one object hold down the `Shift` key and click the drawing objects you want to select.



To cancel a selection, click anywhere in the document outside of the selected drawing object. To cancel the selection of a single object hold down the `Shift` key and click on the object.

To select several drawing objects from one or more diagrams, select the diagrams the objects are attached to. Then choose the **Multi-Select** button from the toolbar and drag a rectangle that encloses all of the drawing objects you want to select.



To move a drawing object, position the cursor on the object and drag the object to the new position. To resize the object, position the cursor on an object handle and drag the handle to its new location. The size of a drawing object will be displayed in the status bar in centimeters while the object is dragged.

When you drag the handle of an object, the object is pulled into alignment with the gridline or the nearest intersection of gridlines (Snap-To-Grid effect). To turn off this effect, hold down the `Ctrl` key while you drag the object. To change the spacing between the gridlines, choose *toolsoptions*.

Position and size of a drawing object can also be specified in the object's dialog box. To open the dialog box, double-click the object.

Tip: To simplify resizing, magnify part of the document. Click the **Zoom In** button and drag to create a rectangle that encloses the part of the page you wish to enlarge.

The mouse wheel can be used to change the zoom factor. Hold down the Ctrl-key and spin the mouse wheel.

5.7.2 Setting Attributes

To change the object's settings, double-click the object to open the dialog box. When the cursor is above the object, it will change from a arrow to a cross. Choose your options in the dialog box and click the *OK* button.

5.7.3 Drawing Order of Objects

The drawing objects will be drawn in the same order in which they were added to the diagram. You can change the layering order within one diagram with the **Bring To Front** and **Send To Back** buttons in the toolbar. If you want to send a drawing behind the diagram, or in front of all other diagrams, you have to cut the object out of the diagram and paste it into the background or into the front diagram.

With the **Bring To Front** and **Send To Back** buttons you can also change the layering order of the diagrams or change the layering order of the datasets within one diagram. To change their order, only one dataset or one diagram should be selected.

5.7.4 Grouping Drawing Objects

Two or more drawing objects of one diagram can be grouped together to form a single unit. Grouped Objects can be treated as a single object. To group drawing objects, select at least two objects and choose the **Group** button from the toolbar. To ungroup a Group Object, choose the **Ungroup** button.

5.8 Formula and Latex Text

If the typesetting program Latex is installed you can type in formulas like the following:

$$\int_0^\infty x^2 e^{-x^2} dx = \frac{\sqrt{\pi}}{4}$$

See also *edit-insert-formula*.

5.9 Text Objects

In the text of an object you can use special characters and the ANSI character sets. Characters or series of characters can be printed in sub- or superscript. Text can contain lines, arrows and symbols.

5.9.1 Subscript and superscript

Subscript can be entered by placing the underscore followed by an opening bracket (_{) before the item you want subscripted, and a closing bracket } after it.

Example:

`a_{123}` will produce

To produce a superscripted character replace the underline with a caret ^ Example: `min^{-1}` will produce

Greek letters can be entered by typing a backslash \ before the corresponding latin letter.

Example:

`\a \w \l \L` will produce

5.9.2 Special characters

Special characters can be produced by using the Character Map program `charmap.exe` or by holding down the **Alt** key and typing the decimal value of the character with a leading 0 using the numeric keypad.

Example 1: To create the Copyright-Symbol ©, hold down the Alt key and type 0169 on the numeric keypad.

Example 2: To create the Register character with the symbol font, type in a backslash \, hold down the ALT key and type in the number 0226 on the numeric keypad. The backslash changes the font to the symbol font.

Example 3: To create the Unicode character "LATIN SMALL LETTER M WITH DOT ABOVE",

- Hold down the Alt key.
- Type the "+"-character (plus) on the numeric keypad.
- Type in the hexadecimal Unicode value, for "m dot" the value 1e41.
- Release the Alt key.

To enable the "Alt+" input mode, add the following key to the Windows registry `HKEY_Current_User\Control Panel\Input Method\EnableHexNumpad` with the value "1" (REG_SZ). You must restart windows if you created a new key.

Beginning with UniPlot R2012.1 Unicode characters can be inserted into a text using the following code `\#xxxx` where `xxxx` is the 4 digit hexadecimal code of the character. For example the "m dot" can be inserted as `\#1e41`.

For most Unicode characters a dot or a double dot does not exists. In this case a combining character can be used, see http://en.wikipedia.org/wiki/Combining_Diacritical_Marks.

For example, a "q dot" can be set as `q\#0307` and a "q double dot" can be set as `q\#0308`.

ANSI Character Codes Chart

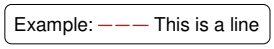
0		32		64	@	96	`	128	€	160		192	À	224	à
1		33	!	65	A	97	a	129		161	¡	193	Á	225	á
2	┐	34	"	66	B	98	b	130	‚	162	¢	194	Â	226	â
3	└	35	#	67	C	99	c	131	*f*	163	£	195	Ã	227	ã
4	┘	36	$	68	D	100	d	132	„	164	¤	196	Ä	228	ä
5	│	37	%	69	E	101	e	133	…	165	¥	197	Å	229	å
6	–	38	&	70	F	102	f	134	†	166	¦	198	Æ	230	æ
7	•	39	'	71	G	103	g	135	‡	167	§	199	Ç	231	ç
8	◘	40	(	72	H	104	h	136	ˆ	168	¨	200	È	232	è
9		41	)	73	I	105	i	137	‰	169	©	201	É	233	é
10		42	*	74	J	106	j	138	Š	170	ª	202	Ê	234	ê
11	♂	43	+	75	K	107	k	139	‹	171	«	203	Ë	235	ë
12	♀	44	,	76	L	108	l	140	Œ	172	¬	204	Ì	236	ì
13		45	-	77	M	109	m	141		173	-	205	Í	237	í
14	♬	46	.	78	N	110	n	142	Ž	174	®	206	Î	238	î
15	☼	47	/	79	O	111	o	143		175	¯	207	Ï	239	ï
16	┼	48	0	80	P	112	p	144		176	°	208	Ð	240	ð
17	◄	49	1	81	Q	113	q	145	‘	177	±	209	Ñ	241	ñ
18	↕	50	2	82	R	114	r	146	’	178	²	210	Ò	242	ò
19	‼	51	3	83	S	115	s	147	“	179	³	211	Ó	243	ó
20	¶	52	4	84	T	116	t	148	”	180	´	212	Ô	244	ô
21	┴	53	5	85	U	117	u	149	•	181	µ	213	Õ	245	õ
22	┬	54	6	86	V	118	v	150	–	182	¶	214	Ö	246	ö
23	┤	55	7	87	W	119	w	151	—	183	·	215	×	247	÷
24	↑	56	8	88	X	120	x	152	˜	184	¸	216	Ø	248	ø
25	├	57	9	89	Y	121	y	153	™	185	¹	217	Ù	249	ù
26	→	58	:	90	Z	122	z	154	š	186	º	218	Ú	250	ú
27	←	59	;	91	[	123		155	›	187	»	219	Û	251	û
28		60	<	92	\	124	|	156	œ	188	¼	220	Ü	252	ü
29		61	=	93	]	125		157		189	½	221	Ý	253	ý
30		62	>	94	^	126	~	158	ž	190	¾	222	Þ	254	þ
31		63	?	95	_	127		159	Ÿ	191	¿	223	ß	255	ÿ

ANSI Character Codes Chart (Symbol Character Set)

0		32		64	@	96	‾	128		160	€	192	ℵ	224	◊
1		33	!	65	A	97	α	129		161	ϒ	193	ℑ	225	⟨
2	┐	34	∀	66	B	98	β	130		162	′	194	ℜ	226	®
3	└	35	}67		X	99	χ	131		163	≤	195	℘	227	©
4	┘	36	∃	68	Δ	100	δ	132		164	⁄	196	⊗	228	™
5	│	37	%	69	E	101	ε	133		165	∞	197	⊕	229	∑
6	–	38	&	70	Φ	102	ϕ	134		166	*f*	198	∅	230	⎛
7	•	39	ə	71	Γ	103	γ	135		167	♣	199	∩	231	⎜
8	◘	40	(	72	H	104	η	136		168	♦	200	∪	232	⎝
9		41	)	73	I	105	ι	137		169	♥	201	⊃	233	⎡
10		42	*	74	ϑ	106	φ	138		170	♠	202	⊇	234	⎢
11	♂	43	+	75	K	107	κ	139		171	↔	203	⊄	235	⎣
12	□	44	,	76	Λ	108	λ	140		172	←	204	⊂	236	⎧
13		45	–	77	M	109	μ	141		173	↑	205	⊆	237	⎨
14	♬	46	.	78	N	110	ν	142		174	→	206	∈	238	⎩
15	☼	47	/	79	O	111	o	143		175	↓	207	∉	239	⎪
16	┼	48	0	80	Π	112	π	144		176	°	208	∠	240	□
17	◄	49	1	81	Θ	113	θ	145		177	±	209	∇	241	⟩
18	↕	50	2	82	P	114	ρ	146		178	″	210	®	242	∫
19	‼	51	3	83	Σ	115	σ	147		179	≥	211	©	243	⌠
20	¶	52	4	84	T	116	τ	148		180	×	212	™	244	⌡
21	┴	53	5	85	Y	117	υ	149		181	∝	213	∏	245	⎮
22	┬	54	6	86	ς	118	ϖ	150		182	∂	214	√	246	⎷
23	┤	55	7	87	Ω	119	ω	151		183	•	215	·	247	⎟
24	↑	56	8	88	Ξ	120	ξ	152		184	÷	216	¬	248	⎠
25	├	57	9	89	Ψ	121	ψ	153		185	≠	217	∧	249	⎤
26	→	58	:	90	Z	122	ζ	154		186	≡	218	∨	250	⎥
27	←	59	;	91	[	123	{	155		187	≈	219	⇔	251	⎦
28		60	<	92	\	124	|	156		188	…	220	⇐	252	⎫
29		61	=	93	]	125	}	157		189	⏐	221	⇑	253	⎬
30		62	>	94	⊥	126	~	158		190	—	222	⇒	254	⎭
31		63	?	95	_	127		159		191	↵	223	⇓	255	□

For legends and similar objects, lines, arrows and symbols can be inserted into the text. To add a line to a string in a text object, open the **Text Object** dialog box and add the following text:

```
Example:  @l{1.1, 1, 3, 255, 0, 0} This is a line
```

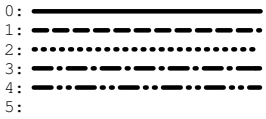


The elements in the example above have the following meaning:

```
@l{L, S, W, RC, GC, BC}
```

Value	Meaning
L	Length of the line in centimeters
S	Line style (0 to 5)
W	Line width (thickness) in steps of 0.1 mm
RC, GC, BC	red-, green-, blue-component of the line color (0 to 255)

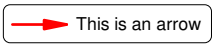
Available line styles:



5.9.3 Arrows in Text

To insert a red arrow into the text, type in the following:

```
@a{1.2, 0, 5, 255, 0, 0, 1, 0.5, 0.2} This is an arrow
```



The elements have the following meaning:

```
@a{L, S, W, RC, GC, BC, P, AL, AW}
```

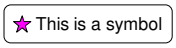
Value	Meaning
L	Length of the line in centimeters
S	Line style (0 to 5)
W	Line width (thickness) in steps of 0.1 mm
RC, GC, BC	red-, green-, blue-component of the line color (0 to 255)
P	Arrow position, (0 = left, 1 = right, 2 = left and right)
AL	Arrow length in centimeters
AW	Arrow width in centimeters

Arrow positions:

5.9.4 Symbols in Text

To insert a symbol into a text line, type in the following:

```
@m{0.4,12,1,255,0,255,0,0,0} This is a symbol
```

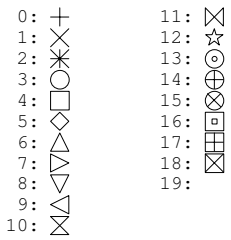


The elements have the following meaning:

```
@m{D, T, W, RF, GF, BF, RL, GL, GL}
```

Value	Meaning
D	Diameter in centimeters
T	Marker style (0 to 18)
W	Line width (thickness) in steps of 0.1 mm
RF, GF, BF	red-, green-, blue-component of the fill color (0 to 255)
RL, GL, BL	red-, green-, blue-component of the line color (0 to 255)

Available markers:

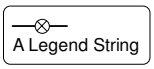


5.9.5 Spaces in Text

Elements can be moved horizontally with the `@s{distance}` command. Positive values for distance move the insert position to the right and negative values will move it to the left. The values are entered in centimeters. E.g. `@s{-0.5}` will move the cursor 0.5 cm to the left.

For example, this command is necessary to create a line element with a centered symbol:

```
@l{1.2,0,4,0,0,0}@s{-0.7}@m{0.3,15,1,255,255,255,0,0,0}@s{0.5} A Legend
String
```

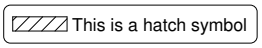


However, you do not have to type in this string. To create a legend for your 1D and 2D datasets, select *diagrammore-diagram-functionscreate-legend-for-2d-datasets*.

For legends and similar objects, rectangles with a hatch filling can be inserted into the text.

Example:

```
@h{1.2, 2, 2, 0.3} This is a hatch symbol
```



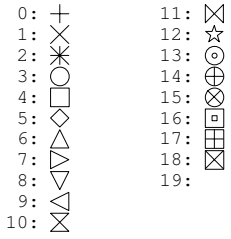
Syntax:

```
@h{L, T, B, LD}
@h{L, T, B, LD, RL, GL, BL}
@h{L, T, B, LD, RL, GL, BL, RF, GF, BF}
@h{L, T, B, LD, RL, GL, BL, RF, GF, BF, RK, GK, BK}
```

The single elements have the following meaning:

Value	Meaning
L	Length of the rectangle (the height is taken from the font).
T	Type (0, 1, 2, 3, ...), 0=No hatch
B	Line width (thickness) in steps of 0.1 mm
LD	Hatch line distance in centimeters.
RL, GL, BL	**red-, green-, blue-component of the line color (0 to 255). (Default** black (0, 0, 0))
RF, GF, BF	**red-, green-, blue-component of the fill color (0 to 255). (Default** transparent)
RK, GK, BK	**red-, green-, blue-component of the edge line color (0 to 255). (D** is black (0, 0, 0))

Available hatch styles:



5.9.6 Tabs in Text

The tab character sequence `@t{size}` can be used to plot text in table form. The number in the braces is the distance of the following text from the left side of the text object in centimeters. In the following example the second column is printed 3 cm and the third column 5 cm from the left edge of the text object.

vollast2.nc	@t{3}LAMABG	@t{5}BEEWG
minimum	@t{3}0.89	@t{5}290.3
maximum	@t{3}0.99	@t{5}325.7
arith. mean	@t{3}0.925455	@t{5}303.1
mean deviation	@t{3}0.0269421	@t{5}8.10909
variance	@t{3}0.00104727	@t{5}115.304
standard deviation	@t{3}0.0323616	@t{5}10.738

vollast2.nc	LAMABG	BEEWG
minimum	0.89	290.3
maximum	0.99	325.7
arith. mean	0.925455	303.1
mean deviation	0.0269421	8.10909
variance	0.00104727	115.304
standard deviation	0.0323616	10.738

Field Functions are special codes that instruct UniPlot to insert information into a document. With fields, you can add and automatically update text, page numbers, legends and other information in a document.

Example: The **documentname** field function inserts the name of the document into a document page. When the document is saved under a new name, the text element is updated.

To insert the field function **documentname**, create a text element and type `@f{documentname}` into the element. The result of the field function will be displayed in the document page, e.g. `c:\test.ipw`.

5.10 Text Placeholder

A placeholder is text that begins with a dollar sign, followed by the name, followed by another dollar sign, e.g. `$Operator$` is a placeholder for a global attribute. For a channel attribute the channel name followed by a period (.) is placed before the attribute name, e.g. `$EngSpd.units$`.

Placeholders can be used in text objects, table objects and diagram titles.

To access an attribute from a specific file, add a file alias to the placeholder, e.g. `$File1:Operator$`. The placeholder `$File2:Operator$` would be replaced by the value of the attribute `Operator` of the second file. Numbering of files aliases starts with 1.

The text must be identical to the attribute name in the NC file.

The placeholder can be typed into the text edit control or the name edit control.

Source for the text:

- *edit-objectplaceholder-dialog*
- *fileimport-data* to load global attribute values
- A channel attribute value from an NC file.
- One data point from a channel of a NC file.

Examples for Placeholders:

Placeholder	Description
`$Operator$`	Global NC-File attribute Operator.
`$Origin$`	Global NC-File attribute Origin. This attribute contains usally the file name of the original data file.
`$Title$`	Global attribut with the name `Title`.

`$speed.units$`	speed channel attribute units.
`$File1:Origin$`	Global attribute Origin from the first data file.
`$(2)File1:Origin$`	The number in parenthesis at the beginning sets the display order in the dialog box.
`$File2:speed.units$`	speed channel attribute units from the second file.
`$speed.1$`	First value of channel `speed`.
`$File3:speed.1;%.1lf$`	First value of channel `speed` from file 3. The value will be displayed with one decimal place, see *printf*.
`$File1:speed.-2; %.2lf$`	Second last data point of `speed` displayed with two decimal places.
`$speed.i();%.2lf$`	Value of a data point. See description below.
`$speed.min();%.2lf$`	Minimum of channel `speed`.
`$speed.max();%.2lf$`	Maximum of channel `speed`.
`$speed.std();%.2lf$`	Standard deviation of channel `speed`.
`$speed.mean();%.2lf$`	Mean value of channel `speed`.

The placeholder `$speed.i();%.2lf$` checks, if the global object _g() contains the following member:

```
_g()._placeholder = [. i = 3]; // Data point 3
```

If the member variable `_placeholder` contains a member variable that is identical with the name behind the period (.), in this example `i`, the value of `i` is the index of the data point. It can be negative to access the last element.

Usage: Before the function *auto_ReplaceTextFromNCFile* is invoked a script function sets the the global element:

```
if (_g().has_key("_placeholder") == FALSE) {
    _g()._placeholder = [.];
}
_g()._placeholder.i = 4;
```

If the placeholder `i` does not exists in `_g()._placeholder`, the function checks if a function with the name ``_placeholder_i exists (prefix `_placeholder_`). Example for the `max` function (placeholder `$n_Mot.max();%.2lf$`):

```
def _placeholder_max()
{
   if (_g().has_key("_placeholder") == 0) {
      return "";
   }
   ncid = _g()._placeholder.ncid;
   varid = _g()._placeholder.varid;
   ssFormat = _g()._placeholder.ssFormat;
   rsMax = max(_NC_vargetEx(ncid, varid));
   return nc_val_format(ncid, varid, ssFormat, rsMax, "");
}
```

To access the channel, the element `_g()._placeholder` contains the member elements `ncid`, `varid`. The member element `ssFormat` can be used to format the string.

The following functions are already defined: min, max, mean, std. You can add your own functions.

5.11 Field Functions

Field functions are especially coded strings in text fields. They can be used to update information in documents automatically. They can be used with table or text drawing objects.

Example: The file name is displayed as a text box in a UniPlot document page. If the document is saved under a new name the field is updated and displays the new name.

Solution: Create a text box and type the following text: `@f{documentname}`. Instead of the string `@f{documentname}` the file name is displayed, e.g. `c:\test.ipw`.

5.11.1 Field Function Syntax

The characters `@f{` designate the beginning of a field function, followed by the field function name, followed by a closing brace `}`. The field function can be followed by a parameter list. The parameters are enclosed by parentheses `()` and separated by a comma. The comma sign is not a valid parameter character.

After the field function is first evaluated, the field result will be displayed. If you double-click the text element, the field result is placed in braces behind the field function.

Example: `@f{date}{11/10/1998}`

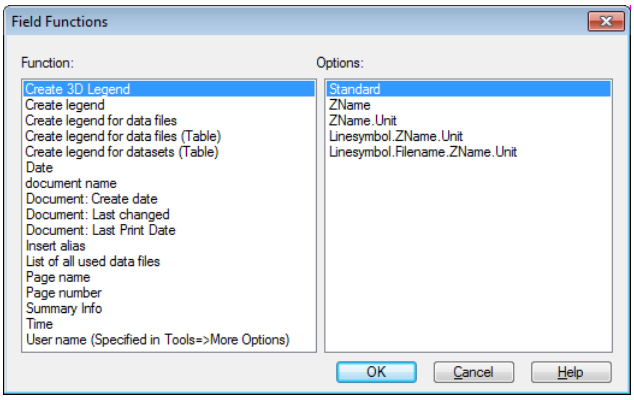
Fields are automatically updated for some user actions such as. When a document page is opened or closed, a text object is edited or when the data configuration is changed. To force a field function to update its field result, use the shortcut key **F9** or choose *editupdate-fields*.

To enable the text object to adjust its size to the text extension the **Automatic Size Update** option should be selected (double-click the text object and set the option in the **Position and Size** dialog page).

Case for field functions is **not** significant. `@f{date}` and `@f{DATE}` are valid field functions.

5.11.2 Inserting Field Functions

To insert a field function into the active document page choose *diagraminsert-field-function* to open the following dialog box:



You can also insert a field function manually as follows:

- Click the **Draw Text** button [A]. Drag to specify the size of the text element.

- Double-click the text element. The text object dialog box will be displayed. Type in the following text: `@f{date}`
- Choose the *OK* button.

5.11.3 Writing user-defined Field Functions

You can write your own field functions. The following example shows the definition of the field function `@f{pagename}`. The name of a field function starts with the prefix **__ff_** (2 underscores followed by **ff** followed by one underscore) (The prefix will be added by UniScript to the field function call). The function name must be specified in small letters. Every field function has 6 parameters. The call parameters which are used in the text element are passed to the field function in the parameter *svParameter*. The function must return the field result as a scalar string. In case of an error, an empty string (`""`) should be returned.

User defined field functions should be saved in a file with the extension `.ic` in the autoload directory. (`uniplot/autoload`). All `ic` files in this directory are loaded at startup.

The parameter *hText* is the handle of the text object which contains the field function. The parameters *hLayer*, *hPage* and *hDoc* are the handles of the parent diagram, parent page and parent document of the text object.

Example:

```
def __ff_pagename(hDoc, hPage, hLayer, hText, nAction, svParameter)
{
    if (hPage != 0) {
        return PageGetTitle(hPage);
    }
    return "";
}
```

You can find more field function examples in the UniScript file `rs_field.ic`.

	Field Functions
@f{3dlegend}	Creates a legend for 3D datasets.
@f{_a}	Displays the text of an alias.
@f{ansi}	Displays an ANSI- or UNICODE character.
@f{area}	Calculates the area for datasets.
@f{channelname}	The field function returns the channelname of the first dataset in the diagram. The field function can be used in a text or table object or in diagram axis title. The text or table object must be assigned to a diagram with datasets.

Continued on next page

Table 5.5 – continued from previous page

	Field Functions
@f{copytextcolortocurve}	Copy the textcolor of the y-Title text to all 1D and 2D curves. The field function must be located in the y-Title-Text
@f{createdate}	Displays the date when the document was created.
@f{cursor_value}	Displays the x- or y-value of a 2D dataset at the cursor position.
@f{datafilenames}	Writes a list of all used data file names used in the page.
@f{datasetfilter}	Displayes the record filter for the given dataset.
@f{date}	Writes the current date.
@f{documentname}	Writes the document file name.
@f{file}	Writes the contents of a text file (maxium 100 lines).
@f{labeldatapoint}	Labels the data points of the original data with its y value.
@f{labeldataset}	The field functions labels a curve with its legend text, set with *XYSetLegendText* or the y-channelname.
@f{labelsinglepoint}	Labels a single data point.
@f{lastprintdate}	Displays the date when the document was last printed.
@f{lastsavedate}	Displays the date when the document was last saved.
@f{legend_table}	Creates a legend table for the data files used in the page or document.
@f{legendchannel}	Creates a legend with the y-channel names of all curves in the diagram.
@f{legendfile}	Creates a legend for all datasets in the parent layer of the text object.
@f{legendstyle}	Creates a legend for a dataset without the legend text.
@f{legendtext}	Displays the legend text of the specified dataset.
@f{legend}	Creates a legend for 1D and 2D datasets.
@f{linkaxisscale}	The function calculates the formula `scale * x + offset` and displays the result as the new axis scale based on a axis given by the diagram name and axis type (x,y,z).
@f{magnify}	...
@f{marker}	This function can be added to the axis title. It displays the used marker of the first dataset.
@f{ncattfromdata}	Displays the specified attribute from a netCDF file (nc file). The function finds the file name via a specified dataset name.
@f{ncattribute}	Displays a netCDF file attribute (nc file).
@f{ncvaluefromdata}	Displays the value of a data point of a netCDF data file channel. The file name of the netCDF data file will be received via the connected dataset.
@f{pagename}	Displays the page name.
@f{pagenumber}	Displays the page number.
@f{revnum}	Displays the document version number.
@f{scaleaxis}	Creates an axisscale depending on another diagram axis.

Continued on next page

Table 5.5 – continued from previous page

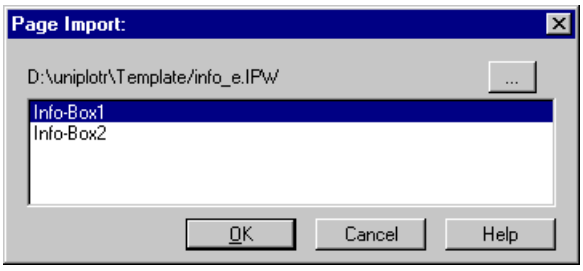
	Field Functions
@f{summaryinfo}	Displays an item from the summary info dialog *filesummary-info*.
@f{symbol}	Displays a character from the symbol character set.
@f{time}	Writes the current time.
@f{update3dcolorlegendtable}	Color legend of a x/y/z dataset. The field function must be inserted into the name of a table object.
@f{update3dcolorlegend}	Color legend of a x/y/z dataset.
@f{xyinterpol}	Displays the interpolated y-coordinate for a given x-coordinate or the x-coordinate for a given y-coordinate of a 1D or 2D dataset.
@f{xyminmax}	Displays the minimum/maximum coordinates of a dataset.
@f{xystatistic}	Displays the following values of a 2D dataset: Minimum, maximum, arith. mean value, mean deviation, variance and standard deviation.
@f{xystatvalue}	Display statistical values for the specified 1D or 2D dataset.
@f{xyvalue}	Displays the x- and y-coordinate of a given 1D or 2D dataset.
@f{xyzstatistic}	Displays statistical information about a 3D map (see example).
@f{xyzstatvalue}	Display statistical values for the specified 3D datasets created from data triples.
@f{xyzvolume}	Calculates the volume, area or the quotient volume/area for a given 3D dataset.

5.12 Information/Legend Boxes

The command *diagrammore-diagram-functionsimport-page* adds the elements of a UniPlot document page to the active page. This function can be used to add standard elements to a page like logos, information legends, etc.

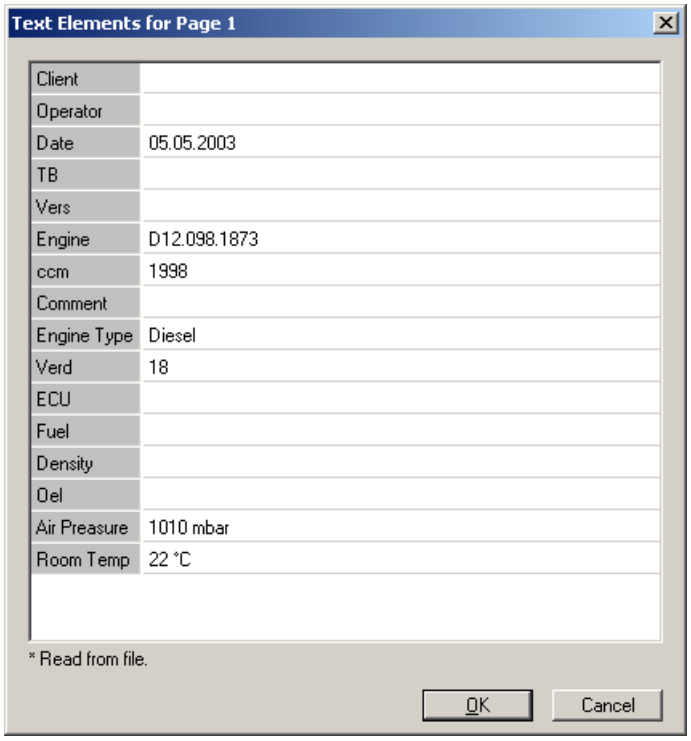
To import a page

- Choose *diagrammore-diagram-functionsimport-page*.
- The following dialog box will be displayed containing the name of the specified IPW file and the page names of this file.



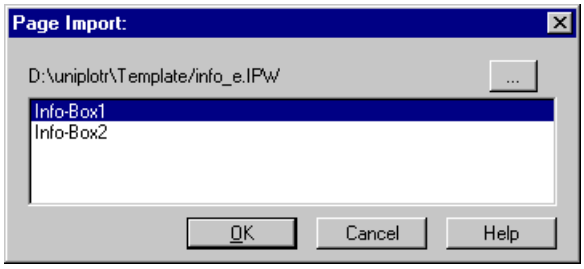
- If the imported page contains a text box, a dialog box will be displayed for text input.

  Example:



- Type in the information and click *OK*. The input will be inserted into the text elements
- If you double-click one text element, the text dialog box will be displayed and you can edit the information.

Pages can contain text legends, frames, diagrams, curves, logos, OLE objects etc. Pages should be created separately for portrait and landscape format. The example file `\UniPlot\Template\Inofo_e.ipw` should not be edited. If you would like to create your own pages or alter the example file, save it under a new name. The example file will be overwritten, if you install a UniPlot update.



To create a new text box

- Choose *filenew*. Choose an empty page.
- Create one text object for every text element.

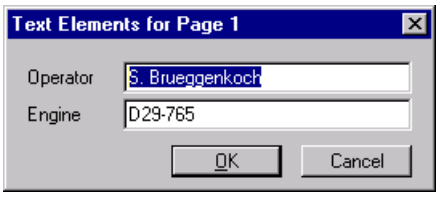
  In the following example, we will create an info box with two elements `"Operator"` and `"Engine"`. Create four text objects with the text `"Operator:"`, `"$(1)Operator$"`, ‘‘ “Engine:”‘‘ and `"$(2)Engine$"`. Then add the rectangle and the horizontal and vertical lines.

Operator:	$(1)Operator$
Engine:	$(2)Engine$

- In the next step, we will group all the elements to one object. To do so, select all the drawing objects. Click one object and then choose the **Multi-Select** button . Drag a rectangle that encloses all drawing objects of this page. Then click the **Group** button .
- Save the new page (*filesave-as*).
- Close the document (*fileclose*).

To test the new info box

- Create a new document (*filenew*).
- Choose *diagrammore-diagram-functionsimport-page*.
- In the dialog box that appears choose the button `"..."`. This will open a **File Open** dialog box. Select the file that contains the new page.
- The file will be opened and the page names will be displayed. Choose the new page name and click *OK*.
- The page will be imported into the active document page and the following dialog box will be be displayed.



- Type in the text and click *OK*.

The place-holder text should be enclosed in dollar signs (`$`) The first dollar sign can be followed by a number enclosed in parentheses `()` (see example above). The number can be used to set an order for the text dialog. The place-holder with the smallest number will be displayed in the upper left corner of the dialog box.

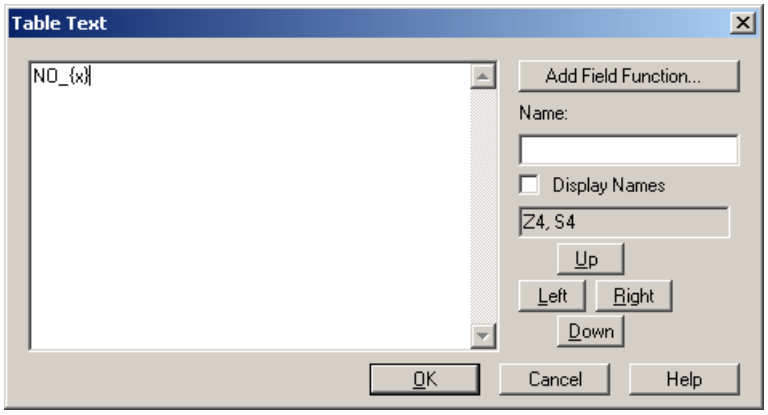
Place-holders will be replaced by the text. Place-holder text will be copied into the name of the text object. When the text receives a double-click, UniPlot checks if the text object belongs to a grouped object. If this is the case, UniPlot checks if the name of the text object contains dollar signs. If so, the text dialog box will be displayed.

5.13 Table Objects

A table object can display text and numbers in columns without using tabulators or blank characters. To create a table click the button from the drawing object toolbar and drag a rectangle with the mouse where you wish to place the table. When you release the mouse button a dialog box will appear in which you can specify the number of columns and rows.

The number of rows and columns is not limited, but if you use the dialog box the number is limited to 50 columns and 50 rows.

To edit the cell text double-click a cell. To navigate to a different cell use the top, bottom, right and left buttons.



The cell name can contain field functions. Field functions can also be placed in the cell text, e.g the date function @f{date} to display the current date.

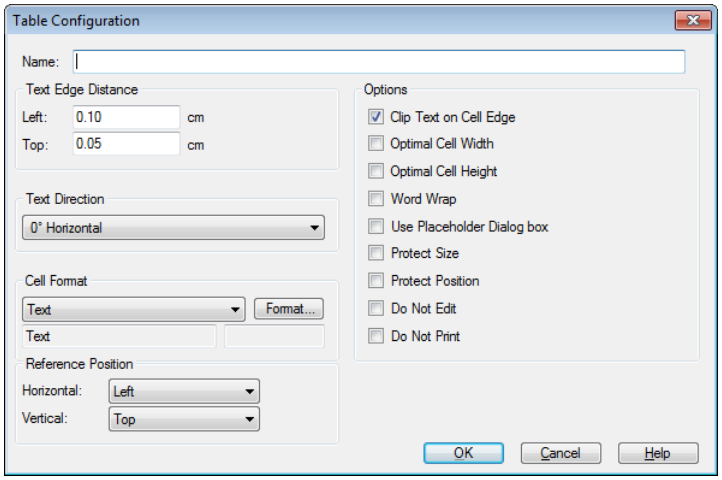
Each cell has a name field that can hold a placeholder. The placeholder can be used to be replaced by attributes from a data file. The placeholder value can be edited in a special dialog box.

The cell text can only be printed with one font, i.e. the text can only be displayed in one color with one font size etc. Each cell can use a different font. The text can be rotated in steps of 90 degrees. Cells can be transparent or opaque. The fill color can be set. The cell border line color and width can be specified. Neighboring cells can be merged.

The text alignment can be set.

The font and font color for the cell text can be set using the format toolbar

More configuration options can be set in the dialog box Table Settings. To open the dialog. select one or more cells and right click one of the selected cells. Choose the **Configuration** command from the context menu. The settings apply to the selected cells.



The context menu offers more commands to add or remove columns and rows and copy the contents to the clipboard, etc.

Cell width and height can be edited using the mouse. By holding down the `Ctrl` key cell width can be set without changing table width.

To drag the table on the page, grab the table on one of the yellow handles.

CHAPTER 6

The Clipboard

6.1 Insert Objects into UniPlot

OLE (Object Linking and Embedding) enables you to insert objects into UniPlot that were created in other applications.

An object can be inserted into a UniPlot document page with the command *editinsert-new-object* or by copying the object using the clipboard.

To insert an object into a UniPlot document page using the clipboard:

- Start the application in which you want to create the object, or open an existing file from which you want to select objects to embed into the UniPlot document.
- Select the object you want to copy.
- Choose **Edit=>Copy** in the application to copy the selection into the Windows clipboard.
- Switch to UniPlot (key: `Alt+Tab`).
- In UniPlot select the diagram into which you want to insert the object and then choose *editpaste*.

To embed an object:

- Select the diagram in which you want to insert the object and then choose *editinsert-new-object*.
- Choose an application in which you want to create the object from the list box.
- Create the object or choose **From File** and select the file which you want to insert into your UniPlot document page.

6.2 Editing an Embedded Object

Embedded objects can be moved, resized and edited without leaving UniPlot. To edit an embedded object, double-click it to open the application in which the object was created.

If the object is activated in place, you can return to UniPlot by clicking outside the object or by pressing the ESC key.

6.3 Copying Drawing Objects

Drawing objects can be copied, cut and pasted between different diagrams of one UniPlot document or between different UniPlot documents.

Note: When an UniPlot drawing object is copied from the clipboard into the same diagram, it will cover the original object. The position and size of the inserted object will remain the same. The copy will be placed at exactly the same position even if inserted into a different document or diagram.If you want to insert drawing objects into another application, choose the *editcopy-page* function.

6.4 Copying Datasets

To copy the selected dataset into the clipboard, choose *editcopy-dataset*.

To insert the datset into a diagram, select the diagram and then choose *editpaste*. The diagram axes will be rescaled automatically.

6.5 Copy Diagrams

To copy a diagram with all of its drawing objects and datasets, choose *editcopy*.

Note: If you want to insert an UniPlot diagram into another application, choose the *editcopy-page* function.

6.6 Copying Pages

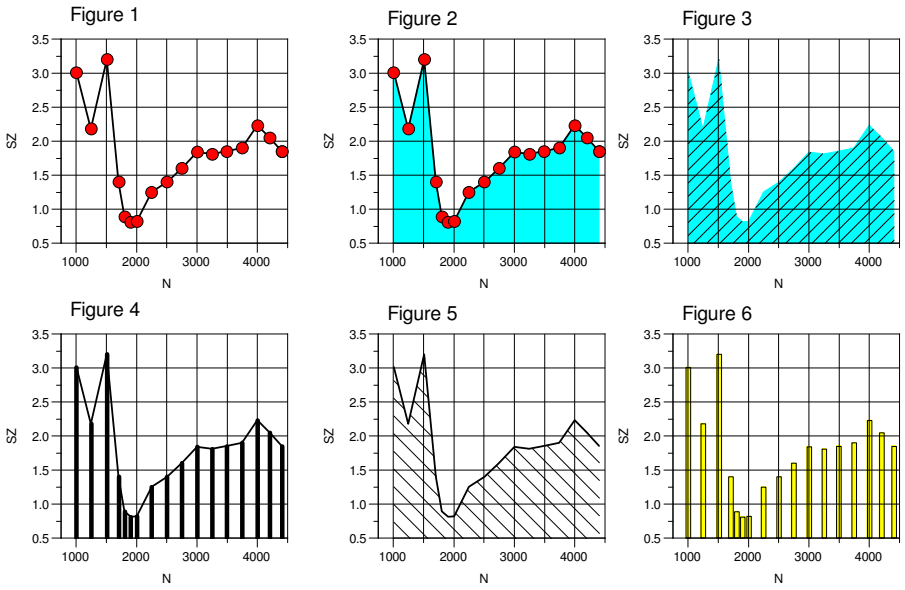
The *editcopy-page* copies the whole page into the clipboard in UniPlot format and as a Windows Metafile.

To paste an UniPlot document into another application, choose *editpaste* in the application into which you want to insert the UniPlot diagram.

CHAPTER 7

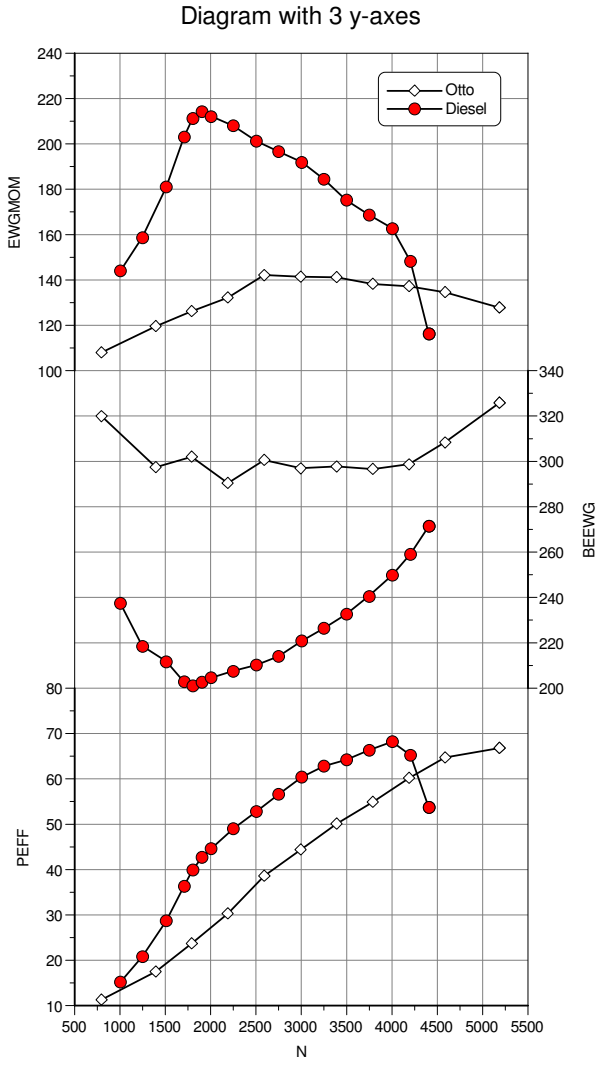
Examples

7.1 2D Dataset Attributes



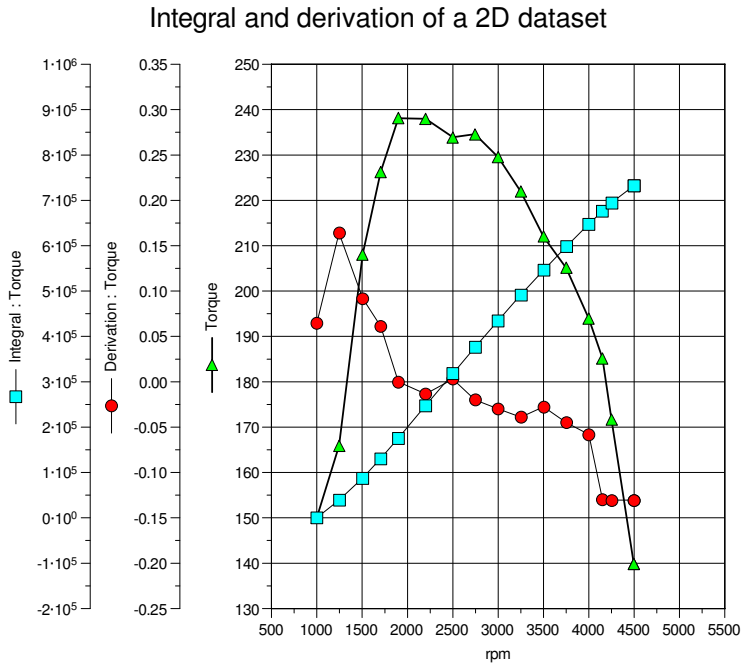
- Click the curve to select the dataset and then choose the **2D Curve** button from the toolbar, or
- Double-click the curve, or
- Choose the **Dataset List** button from the toolbar, select the name of the dataset and then choose the Configuration button to open the 2D Curve dialog box.

7.2 Diagram with 3 Overlapping y-axes



- Choose *filenew* and select the diagram with the multiple y-axes option. In the following dialog box, choose **3 y-axes** and then click *OK*. A diagram with one x-axis and three y-axes will be created.
- To import a dataset, select the lower diagram by clicking inside the diagram frame. The selection will be displayed by blue handles at the diagram corners. Choose *fileimport-data*, open the file `VOLLAST.NC`. In the Import dialog box, choose x = `N` and y = `PEFF`. Make sure the **Axes** checkbox is selected and then choose the **Load** button.
- Select another diagram, specify the next dataset and click **Load**.
- To create a square diagram grid, choose *diagrammore-diagram-functions* and then select **Scale Axes for Square Grid**. Click **OK** to execute the function.

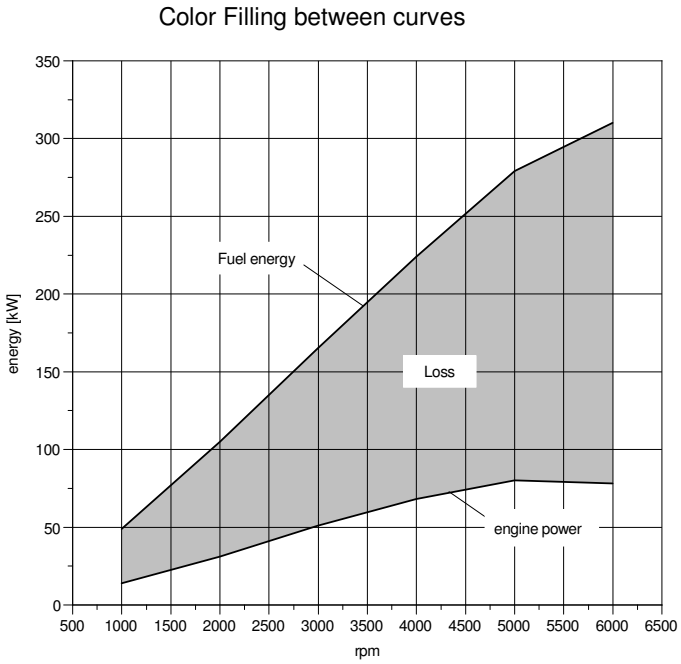
7.3 2D Dataset - Derivation and Integration



- Import the x = `N` and y = `Mom` 2D dataset from the `vollast.nc` file.
- Choose *datamore-data-functions* and then select the **Create Derivation Dataset** option. A new diagram with a shifted y-axis and a hidden x-axis will be created.
- To create an integral dataset from the original (N, Mom) dataset, select the dataset and then choose *dataintegral*). Again, a new diagram with a shifted y- and a hidden x-axis will be created. However the y-axis will now overlap the y-axis from derivative diagram.
- To shift the axis more to the left, choose *diagramlayout* and type in 4 (cm) for the y-axis distance.
- Choose data point symbols for the three datasets and then create a legend (**Diagram=>More Diagram Functions** and select **Legend for 2D datasets**, **Create Legend for all Diagrams**).
- Double-click the legend. Mark `@l{1.2, ..., 0.475}` and copy (`Ctrl+C`) the selection into the clipboard. Close the dialog box and double-click the Mom axis title. Position the cursor at the beginning of the text and then press `Ctrl+V` to insert the clipboard contents.

Another way to display the marker in the y-title is to insert the field function *@f{marker}*. This function displays the marker from the first dataset in the diagram.

7.4 Diagram with 2 Curves and Color Filling



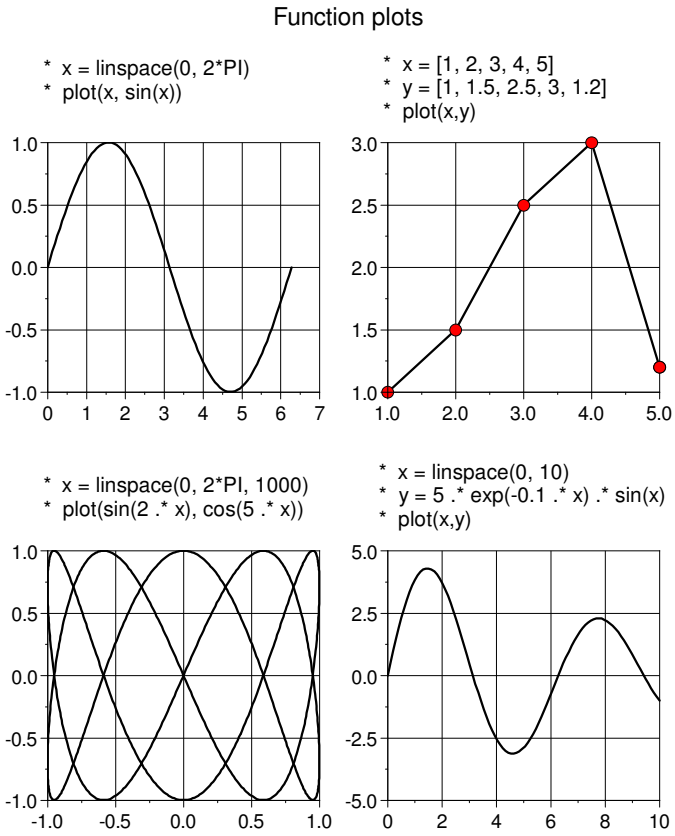
- Load the x = Drehzahl (rmp), y = Brennstoffenergie (fuel energy) 2D dataset from the `ENERGIE.ASC` file in the `UniPlot\Samples` directory.
- Choose the **1D and 2D Dataset Configuration** button from the toolbar to open the 2D Dataset dialog box. Click the **Fill Color** check box and then choose the **OK** button. The area under the curve should be color filled down to the x-axis.
- Load another dataset from the ENERGIE file. Load the x = `Drehzahl`, y = `Nutzleistung` 2D dataset.
- Choose the 2D Dataset button from the toolbar to open the dialog box. Click the **Fill Color** check box. Open the **Fill Color** dialog box and choose the color white, then click the *OK* button. The area under the curve should be color filled down to the x-axis.

Hatch Filling

To fill the area between two curves with a hatch do the following:

- Select the two curves. Click the first curve. To select the second curve hold down the `Shift` key while you click on the second curve.
- Choose *data-more-data-functions* and select **Hatch Fill between two 2D datasets**.

7.5 Plot a Function f(x)

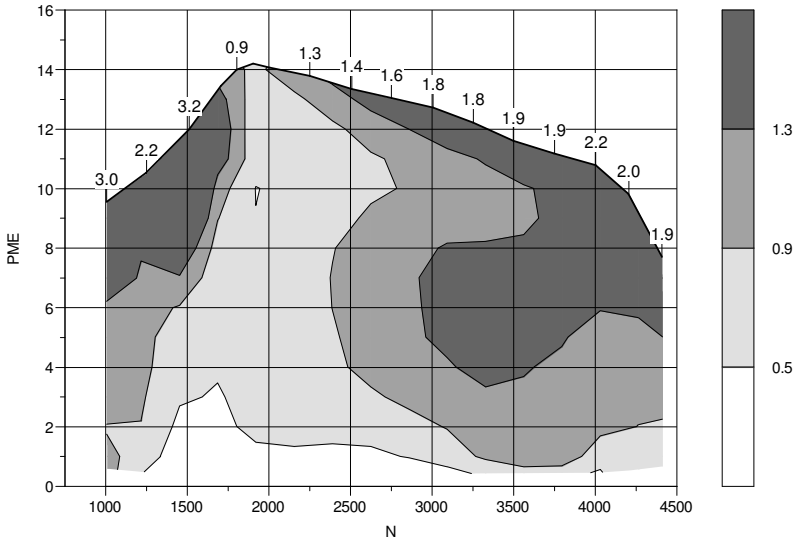

Function plots
* x = linspace(0, 2*PI)
* plot(x, sin(x))
* x = [1, 2, 3, 4, 5]
* y = [1, 1.5, 2.5, 3, 1.2]
* plot(x,y)
* x = linspace(0, 2*PI, 1000)
* plot(sin(2 .* x), cos(5 .* x))
* x = linspace(0, 10)
* y = 5 .* exp(-0.1 .* x) .* sin(x)
* plot(x,y)

To plot one of these functions:

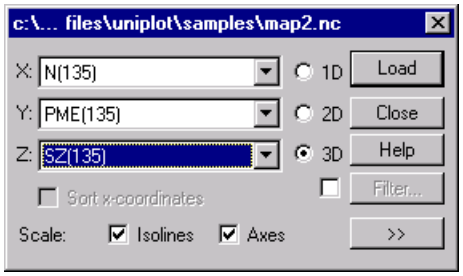
- Choose *view-command-window* or choose the UniScript button in the toolbar to open the UniScript command window.
- Type in the program lines from to create the plot. The *plot* function will create a new document with a diagram and the dataset.

You can find more information about UniScript in the second part of this user manual.

7.6 Contour Diagram with Color Gradient and Legend

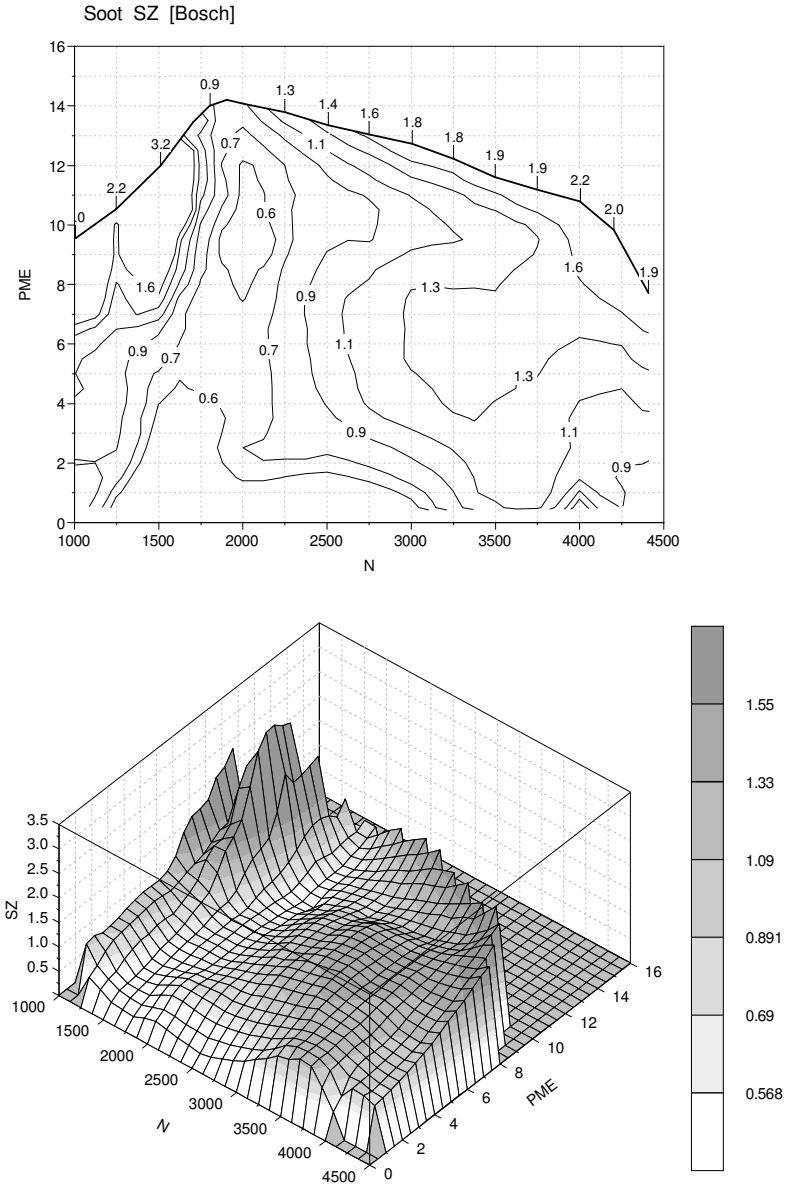


- To open the `map2.nc` data file from the `UniPlot\Samples` directory, choose *file-import-data*. Select the **3D** option button and then select `N `` for the x-axis, ``PME` for the y-axis and `SZ` for the z-axis. Make sure the **Isolines** and the **Axes** check boxes are marked and then choose the Load button.



- Double click an isoline to open the 3D dataset dialog box. Select all the isoline values in the list box and choose the **Delete** button to remove all the isoline values.
- Type 0.5, 0.2, 1.5 into the text box and then choose the Insert button to add the values 0.5, 0.7, 0.9, 1.1, 1.3 and 1.5 to the list box. Select all values including the value **min** and click the Fill Color button to specify the color gradient.
- Choose white (red=255, blue=255, green=255) for Minimum and black for Maximum (the color gradient will be displayed in the left box) and then choose the OK button.
- In the 3D dataset dialog box, choose the **Isoline** tab and select the **Color Filling Between Isolines** check box and then choose the **OK** button.
- To create a color legend, choose *data-more-data-functions* and select the **Color Legend** option. Then click the **OK** button.

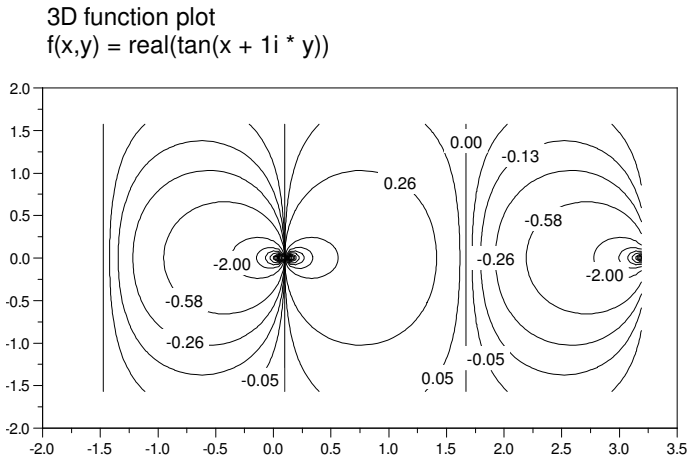
7.7 Contour and 3D Diagram



- Open the MAP2.NC data file and import the x = N, y = PME and z = SZ 3D dataset.
- To create a copy of the diagram, choose *editcopy-diagram*
- To insert the clipboard contents into the document, choose *editpaste*. The pasted diagram will be inserted at the same position as the original diagram.
- Select the copied diagram and move it to the lower area of the document. Choose the **3D** button from the toolbar to change the diagram to a 3D diagram.
- To specify a color gradient for the surface map, choose the **3D Surface** button from the toolbar. Then choose the **3D Surface** tab and select the **Surface with Color Gradient** option.

- To create a color legend, choose *datamore-data-functions*. Select the **Color Legend** option from the list box.

7.8 Plot a Function f(x,y)



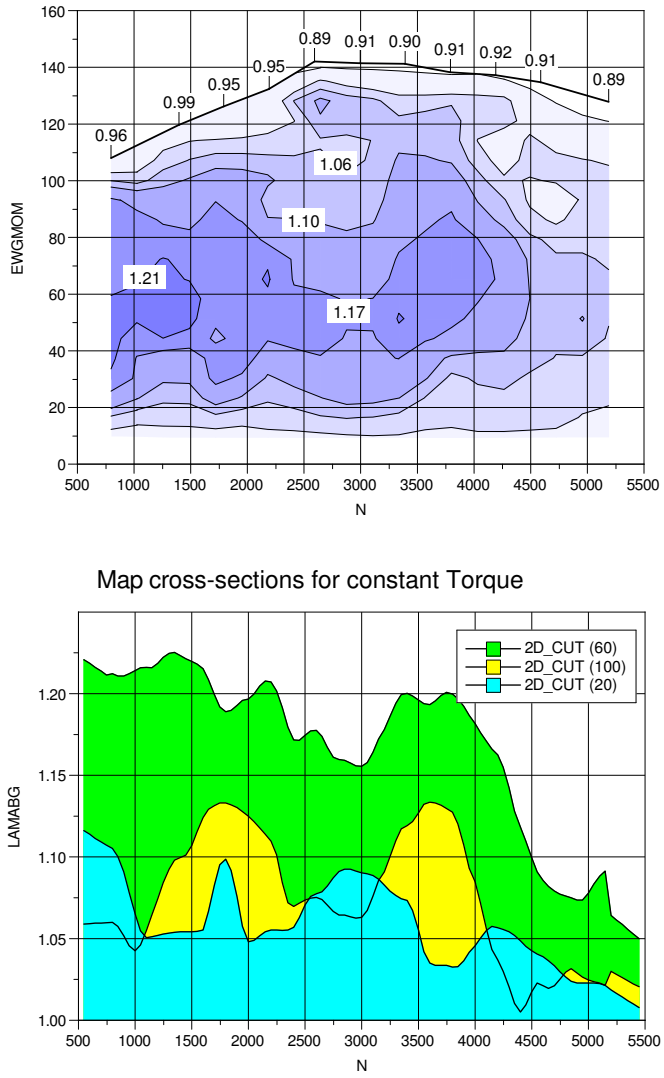
UniScript Program:

```
nRC = 40;
xmin = 0.05-PI/2;
xmax = 0.05+PI;
ymin = -PI/2;
ymax =  PI/2;

x = linspace(xmin, xmax, nRC);
y = linspace(ymin, ymax, nRC);
<xx, yy> = meshdom(x,y);
zz = real(tan(xx + 1i * yy));

def _plotMatrix(Name, zz, rvRange)
{
  hdoc = DocCreate();
  hlayer = LayerCreate();
  DocAddLayer(hdoc, hlayer);
  hdata = XYZCreateFromMatrix(Name, zz, rvRange);
  LayerAddDataset(hlayer, hdata);
  LayerAutoScale(hlayer);
  DocReplot(hdoc);
}

_plotMatrix("real(tan(xx + 1i * y))", zz, [xmin, ymin, xmax, ymax]);
```

- To create a new editor, choose *filenew*. Select the Editor option from the list box to create an empty editor.
- Type in the program code above.
- To execute the new program, choose *uniscript-save-execute*. If you typed in the program correctly, a new document with an isoline diagram will be created. If an error occurs during the compilation process, correct the error and choose **Save/Execute** (**F4**) again.

To receive Help for a function, position the cursor on a function name and press **F1**. More information about using the program language UniScript is provided in second part of the manual.

7.9 Contour Cross-section



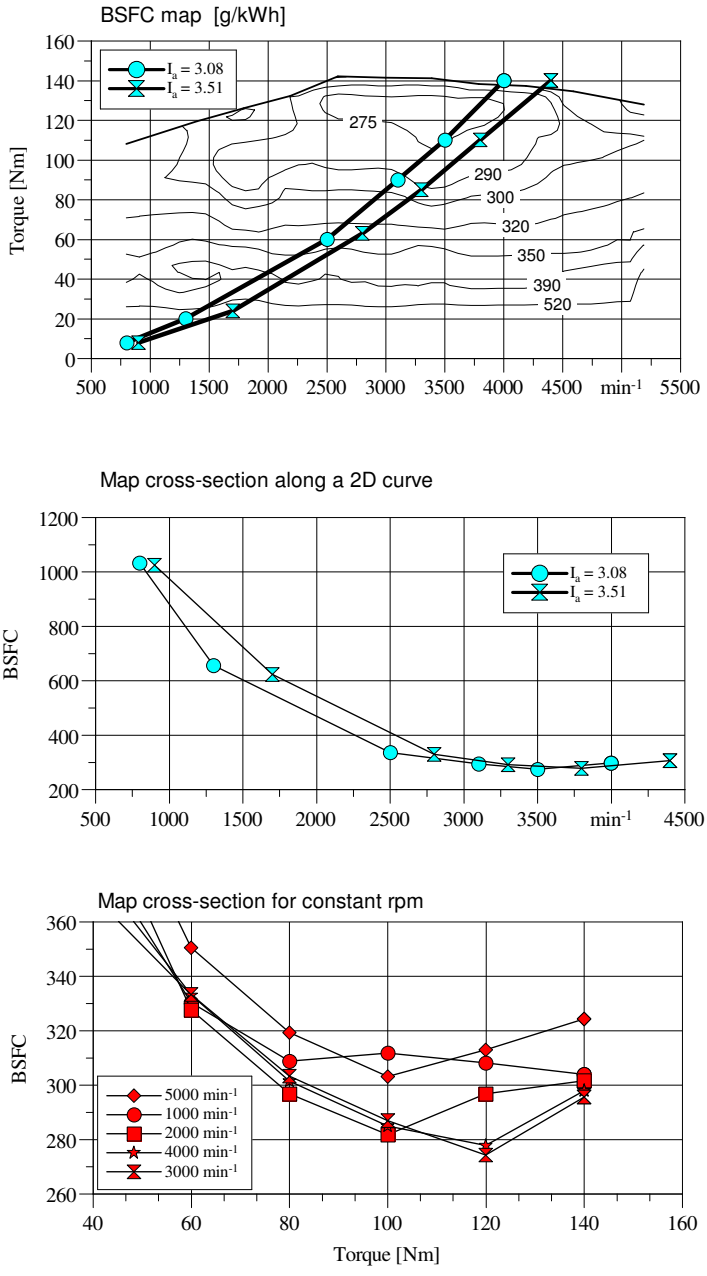
- Import a 3D dataset (x = N, y = EWGMOM, z = LAMAGB) (LAMABG = Lambda in Exhaust) from the `KENNFELD.NC` data file.
- Click one isoline to select the 3D dataset. Choose *datamore-data-functions*. Choose the function **Cross-section for y = constant** from the list box.
- In the following dialog box, some cross-section values will be displayed. Delete all values except 20;60;100 in the edit field. The values must be separated by one semi-colon. No separator should be placed in front of the first and after the last value.

A new document will be created with cross-section lines for the values 20, 60, and 100 Nm.

In the next step, we want to create a copy of the diagram with the contour map into the document.

- Choose *editcopy-diagram*. The diagram, the datasets and the legend will be copied to the clipboard.
- Select the document with the contour map. Choose the name of the document from the Window menu.
- Insert the diagram from the clipboard into the document. Choose *editpaste*.
- Move the diagram to the lower part of the document.

7.10 Cross-section along a 2D Curve



To create a cross-section along a curve in the x/y plane, a 2D dataset is necessary. For every data point, a z-value will be computed from the contour map. The z-values can be plotted as a x/z-, a y/z-curve or the z-values can be plotted over the arc length.

In the example, the cross-section was computed along the driving resistance curve in a fuel consumption map.

To create the cross-section, load the (`UniPlot\Samples\Sample.ipw`) example file and repeat the steps from the previous example.

CHAPTER 8

Importing Data

UniPlot supports data import of the following file formats:

- Text format (ASCII files or CSV files, DSV files)
- Excel (Version 2.1 to Excel 2007)
- netCDF file
- MDF format (some systems which use the MDF file format to store data are: VS100/INCA of ETAS GmbH, Germany, Drive-Recorder of IAV GmbH, Germany, and CANalyzer of Vector Informatik GmbH, Germany, see *import-of-mdf-files*)
- FEVIS, Combustion Anaylzing System of FEV Motorentechnik GmbH, Germany
- CAS, Combustion Anaylzing System of MTS Powertrain Technology Division and FEV Motorentechnik GmbH, Germany
- COMBI, Combustion Anaylzing System of SMETec GmbH, Germany
- IFILE, Combustion Anaylzing System file format of AVL List GmbH, Graz-Austria. See *import-of-ifile-data-files*.
- XONE, Test Bench Format, Schenck Pegasus GmbH
- Puma 5.4, Test Bench Format, AVL List GmbH, Graz-Austria
- WFT, Waveform File of Nicolet Instruments
- UTX-Format (UniPlot Text File format), see *UTX Data File Format*
- dBase-Format
- Lotus-Format
- Tektronix Waveform Format (:file:`*.isf`)
- Adapt Format, Test Bench Files, MTS Powertrain Technology Division (http://www.mtspt.com)
- FAMOS (see *import-of-famos-files*), imc Messsysteme GmbH, (http://www.imc-berlin.de)
- DIADem (see *import-of-diadem-files*)
- ASAM-ODS (Browser for API 3 *asam-ods-browser-(rpc)* and API 4 *asam-ods-browser-(corba)*)

- Import of DIAdem TDM- and. TDMS files (see *import-export-of-tdm-files*).
- *import-of-mdf-stiegele-files* (Stiegele Datensysteme GmbH und Caesar Datensystem GmbH)
- MATLAB-Format 4 and 5 (*import-of-matlab-data-files*).
- MDF4: *import-of-mdf4-files*.
- BLF (CAN): *import-of-can-files*.

If a filter is not loaded into UniPlot choose *tools-add-in-manager* to install the filter. Not all of the above filters are installed with the UniPlot setup program. If you do not find the filter please contact us.

Company-specific binary and text formatted files can be imported using the built in programming interface.

8.1 Load a dataset

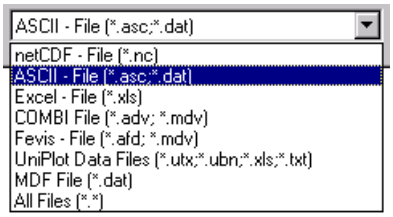
The import process is completed in two steps:

- Load, convert and save the data in a netCDF file.
- Load the data from the netCDF file, create a dataset and attach it to a diagram.

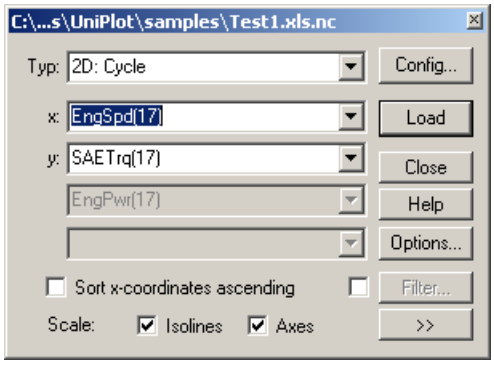
  The first step is only necessary to import the first dataset. It can take anywhere from seconds to several minutes to convert a Text, Excel, or a data file in another format depending on the number of data points in the file. If the data file has multiple columns, they can be directly imported from the netCDF file. The netCDF file will be stored in the directory of the Text or the Excel file under the same name, but with an `.NC` filename extension.

  To import data from a Text or Excel file

- Choose *fileimport-data*.
- In the **File Open** dialog box, select the type of file. Select the name of the data file and click *OK*.



- UniPlot loads the file and creates the netCDF file. If a netCDF file with the same name already exists, you will be asked if you would like to overwrite the file or save it under a different name.
- If the creation of the netCDF was successful, the following dialog box appears:



In this dialog box, you can specify the dataset you want to import.

1D, **2D**, **3D**

Specify the dataset type you want to load.

x, y, z Column

To import the dataset, each axis must be assigned a data column. If the data columns have names above the data and their position was defined correctly in the **Import Options** dialog box, the combo boxes will display the names. Otherwise, the columns will be enumerated (Col1, Col2,...).

Scale Isoline Values

If the **Isoline** check box is selected, isolines will be created for a 3D dataset. The isoline values are chosen to allow them to be spread as evenly as possible over the map.

Scale Axes

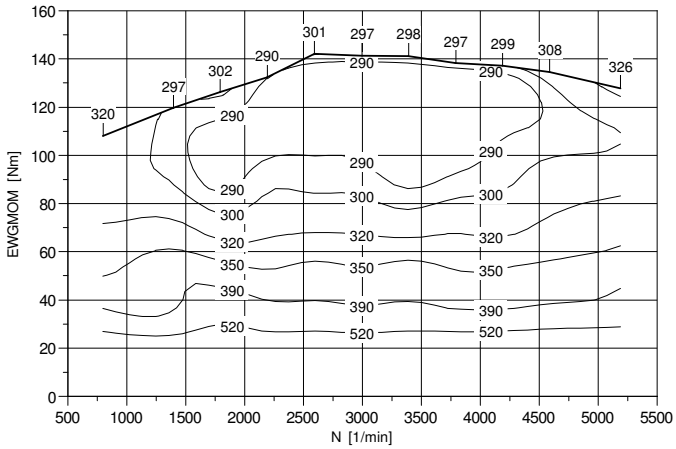
If this check box is selected, the diagram axes will be scaled automatically.

Excerpt from an Text Data File:

```
RPM      Torque   Torque2 bsfc     ... Power    PME
1/min    Nm       Nm      g/kWh    ... kW       bar
1249     55.70    53.51   934.1    ... 7.27     7.05
1243     48.97    47.14   933.2    ... 6.14     6.19
.
.
.
6730     24.27    23.38   239.1    ... 17.68    3.07
6763     15.66    15.09   157.7    ... 11.71    1.98
6740     8.04     7.75    123.6    ... 5.99     1.02
```

The first row of the file contains the names of the data columns. The names are placed two rows before the first data row. In order to display variable names, choose 2 for the column title position in the **Import Options** dialog box.

To create a fuel consumption map from the data excerpt above, choose the **3D** radio button. Select the `N` data column for the x-axis, `MEFF` for the y-axis and `BEEWG` for the z-axis. Choose the **Load** button to import the data into the selected diagram.



Important: In previous UniPlot versions, the Full Load Line (WOT) definition had to be placed in the first data column. In UniPlot 2.0 and further versions, the definition must be located in the last column.

8.2 Format Description for CSV, Text and Excel Files

When using CSV (Comma-Separated Values), Text and Excel files, the following format description must be paid attention to:

The data must be arranged in columns. The first data row must be occur in the first 256 rows.

Number of Columns: The maximum number of columns is limited to 8000.

Number of Rows: The maximum number of data rows is only limited by the amount of available memory. For a 1D or 2D dataset, at least 1 data row is required. A 3D dataset requires 5 data rows.

Column Separator: The import filter accepts the following column separators:

Column Separator	Description
";"	Semi-colon
"\t"	Tab character
","	comma
" "	one or more white spaces

The import filter searches for the column separater in the last 4 rows of the first 256 rows. The separator which occurs most is used. If the separator is a comma the decimal sign must be a period (.). The white space is only used if no other separator is found.

Decimal Separator: Valid decimal separators are the period (.) and a comma (,). The decimal separator is found automatically.

Channel and Unit Names: The file may contain one row with column names (channel names) and one row with unit names. The column names must appear above the unit names. If 80 % of all names are unique, only then will the import filter use them as channel names. If the filter does not find the channel names, the channel name will be created from the name `Col` followed by the index of the column. If a name occurs more than once, the names are enumerated to create unique names.

Column Type: The data is saved in the NC file as real4 (float) values. If the column contains no valid real or integer values, the complete column will be saved as a text channel. If the fields contain valid date or time values e.g. `2007-01-17`, the column will be saved as date or time values (datatype = date or datatype = datetime or datatye = time).

Missing Values: Empty fields or fields containing one of more of the following characters "*", "-", "#" will be saved as missing_values. Fields with the following values will also be saved as missing values (case is ignored): "not a number", "nan", "1.#inf", "no value", "missing*". "#" in the first column will set all values in the row to missing.

Comment Lines: The rows above the data columns may contain comments These rows will be skipped automatically during import.

Double Quotes (""): Fields may always be delimited with double quotes. The delimiters will always be discarded. Numbers in double quotes will be treated as text.

The following example shows a valid file in text format.

```
Speed   Torque  Power
1/min   Nm      kW
1249    55.70   53.51
1243    48.97   47.14
3567    38.54   27.14
6763    15.66   15.09
6740    8.04    7.75
```

All data rows should contain the same number of fields (columns). In the following data file, the 5th row contains only three numbers. The third value of the `Power` channel will be saved as "missing" in the NC file.

```
Speed; Torque; Power; B
rpm; Nm;  kW;  -
1249; 55.70; 53.51;   7.27
1243; 48.97; 47.14;   6.14
6730; 24.27;      ; 239.1
6763; 15.66; 12.03;  15.09
6740; 8.04 ;  7.75;   5.99
```

The following file is a valid CSV file with a date and time column. Text fields are delimited with double quotes. The spaces in the channel names are replaced by underscores in the NC file.

```
"Date","Time","Rate","Num Average","Operator ID","State Label","Sample Label","Engine_upl"
,,V,,,,,""
"27/04/2006","09:59:13.150",1.000000000000000000,1.0,,,,"Otto, Diesel",
"27/04/2006","09:59:13.250",2.000000000000000000,1.1,,,,"Otto",
"27/04/2006","09:59:13.350",3.000000000000000000,1.2,,,,"Otto",
"27/04/2006","09:59:13.450",4.000000000000000000,1.3,,,,"Otto",
```

8.3 UTX Data File Format

The following chapter will describe an easy to create and easy to read file format for UniPlot (UTX Format).

Pros

- Easy to create with all programming languages.
- The file can be created and edited manually with a text editor or spread sheet program.
- Files that are already organized in column form can easily transformed into this format.

Cons

- All channels (columns) must have the same number of data points.

8.3.1 Format Decription

The file consist of two sections, the description block (header) and the data block.

The file starts with the description block. This block begins with the keyword `UXX-BEGIN` (case will be ignored) and ends with the keyword `UXX-END`.

The description block is followed by the data block. This section consists of the data organized in columns in text form (ASCII).

Example 1: The following example file contains two channels: The channel `Speed` [rpm] in the first column and channel the `Torq` (Brake Torque) in the second column. The columns are separated by a tab sign.

The description section contains three attributes: `Operator`, `Channelname` and `unit`. `Channelname` and `unit` are so called channel attributes, i.e. for every channel in the file the attribute holds one value. The value `$1` for the channel name defines that the channel names are found one line after the description block. `Channelname` has the two values `Speed` and `Torq` and the attribute unit holds the values `rpm` and `Nm`. The attribute `Operator` is a global attribute with the value `"Peter Miller"`.

```
UXX-BEGIN
Operator = Peter Miller
Channelname =  $1
unit = $2
UXX-END
Speed       Torq
rpm         Nm
528,000     96,00
474,000     76,00
456,000     52,00
```

Example 2: UniPlot can also read the file if it was created with the spread sheet program Excel and saved as an Excel file (all formats inclusive Excel 2007). In this case the keyword `UXX-BEGIN` should be in cell `A1`, the attribute name should be located in first column and the attribute value in the second column.

	A	B
1	UXX-BEGIN	
2	Bearbeiter=	Peter Müller
3	Kanalname=	$1
4	Einheit=	$2
5	UXX-END	
6	N	Be
7	1/min	G/kWh
8	528,0000	1096,0000
9	474,0000	1076,0000
10	456,0000	1052,0000

Example 3: Data arranged in rows can be imported. In this case the attribute `uxx-transposed = 1` must be specified.

```
UXX-BEGIN
columnseparator = " "
Operator = Peter Miller
Channelname =  $1
unit = $2
uxx-transposed = 1
UXX-END
Speed    1/min   528,000  474,300  456,000
Torq     Nm       96,00    76,00    52,00
```

8.3.2 The Description Block

The description block holds the global attributes (with a scope for the hole file) and the channel attributes.

8.3.3 Global Attribute

A globale attribute has the same syntax as an assignment:

```
NAME = VALUE
```

or

```
NAME = VALUE-LIST
```

The attribute name `NAME` can be a string of up 40 characters. It can start with an underscore `"_"` or a character followed by up to 39 characters, underscores of digits. Special characters should not be used for attribute names (they will be replaced by underscores). The character sequences `UXX-BEGIN` and `UXX-END` are invalid names. `VALUE` can be an number (integer), a real number or a character string.

Integer

An integer is a number of the form 123 or -123. If the number contains an decimal separator it will be saved as a real number. E.g. 1.23e2 (= 123) will be saved as a real.

Real

A real is a number of the form 12.3e1, 123. or 123.0. The decimal separator is a dot or a comma.

String

If the attribute value is neither an integer or a real number it will be saved as a string value. The ANSI character set should be used for strings (see table in the appendix).

Example:

```
Date=      12.06.1998
Engine=      12AT29
```

Strings can be enclosed in double quotes.

VALUE-LIST is a sequence of values of one data type (integer, real or string), which are separated by separator sign (default is a tab character).

Example:

```
Parameter= 12  24  22,5
Range= 1  24576
```

8.3.4 Channel attributes

The values for the channel attributes are organized as a list enclosed in square brackets [ ]. They are separated by the separator sign (default is a tab character).

```
NAME = [VALUE VALUE VALUE]
```

The list must contain a value for each channel. The order must be identical to the column order in the data block. All values must have the same data type, i.e. numbers and strings should not be mixed.

Instead of assinging a list of values in the form:

```
NAME = [ VALUE-LIST ]
```

the channel attribute can be specified in the following form:

```
NAME = $rownumber
```

In this case the attribute value can be found in row rownumber after the keyword UXX-END. This makes it easier to edit the data with a spread sheet program.

Continuation Line

The attribute list can be written over multiple lines. Te continution character "&" at the end will indicate if the list will be continued in the next line. The "&" character should only appear on the left side of the equal sign (=) at the end of the line.

Example:

```
Channelname = [Speed  &
               Torq  &
               BHP]
```

is the same as

```
Channelname = [Speed Torq BHP]
```

Continuation lines are only valid in the description block and not in the data block.

Comments

A comment is a sequence of characters beginning with # in the first column. With UniPlot 5.12.0 a data row can be marked as a comment row by inserting the character "#" at the beginning of the row. The values in a comment row will be saved in the NC file as missing_values.

8.3.5 The Data Block

The data block is a text block in ASCII format with an unlimited number of rows and up to 8000 columns.

The default column separator is one tab character. If a different column separator should be used it must be specified in the description block, e.g. `Columnseparator = ";"`.

Continuation lines and comments are not allowed inside the data block. Empty lines will be ignored.

Channels with Date/Time data can be specified in the form `"25.01.1996 8:30:00"`, `"25.01.1996"` (only date) or `"8:30:00"` (only time).

8.3.6 The standard attributes

The following standard attribute names are reserved and have a defined meaning:

Global Attributs

Columnseparator
Scheme
uxx-transposed

Channel Attributes

Channelname
Unit
Datatype

The standard attributes are not case sensitive, i.e. Datatype, datatype, DATATYPE are the same attribute.

8.3.7 uxx-transposed

If the data is arranged in rows instead of columns the file must contain the attribute `uxx-transposed = 1`. If the data is arranged in columns the attribute is optional.

8.3.8 Channelname

This attribute is mandatory.

The rules for channel names are the same as for attribute names. The channel names can be a string of up 40 characters. It can start with an underscore `"_"` or a character followed by up to 39 characters, underscores of digits. Special characters should not be used for attribute names (they will be replaced by underscores). The character sequences `UXX-BEGIN` and `UXX-END` are invalid names.

```
Channelname=    [Speed  Tory BHP]
```

Instead of the this form you can specify the channelnames as following:

```
Channelname = $rownumber
```

In this case the channelnames can be found in row `rownumber` after the keyword `UXX-END`. This makes it easier to edit the data with a spread sheet program.

If the channel names are given in the form `Name [unit]` and no other units are specified the name will be split into channel name and unit. Example: `Torq [Nm]` will be split in the channel name `Torq` and the unit `Nm`.

With the following command the splitting can be disabled. Type the command into the command window:

```
WriteProfileInt("Settings", "UTX_Split_Name_Unit", 0)
```

8.3.9 Columnseparator

The column separator separates the contents of cells within a row. The default column separator is one tab character. If a different column separator should be used it must be specified in the description block, e.g. `Columnseparator = ";"`. The following separtors are supported:

Character	Description
"\t"	(Default value) A single tab character (backslash + t)
" "	Multiple spaces and tab characters
"\b"	A single space (backslash + b)
";"	a semi-colon
","	a comma
"x"	a single character

```
Columnseparator=    ";"
```

8.3.10 Unit

Unit of a channel.

Example:

```
Unit = [rpm  Nm  g/kWh]
```

or

```
Unit = $rownumber
```

8.3.11 Datatype

The `Datatype` attribute specifies the channel data type. If the attribute is not specified all channels will be saved as `"real4"`. The following data types are valid:

Datatype	Description

int1	One byte integer signed
uint1	One byte integer unsigned
int2	Two byte integer signed
uint2	Two byte integer unsigned
int4	Four byte integer signed
uint4	Four byte integer unsigned
real4	Floating-point numbers (4 Bytes)
real8	Double precision floating-point numbers (8 Bytes)
stringNNN	character strings with length definition. NNN stand for an integer in the range 1 to 255, e.g. string80
date	Date (8 Bytes)
time	Time (8 Bytes)
datetime	Date and time (8 Bytes)

```
Datentyp= [date  real4  string12]
```

The list must not contain a datatype value for every column. If the number of elements is smaller than the number of channels, the datatype is set to `"real"`.

8.3.12 Scheme

Scheme is a character string created from the company name and/or the program name which created the file as well as a version number.

Example: (Company FVM, Test Cell Software P13, Version 1)

```
Scheme= FVM P13-1
```

8.3.13 Creating an UTX data file

To store your data in the utx format

- create a file with the file name extension `.UTX` or alternatively `.TXT`.
- Write into the first line the keyword UXX-BEGIN followed by new line separator (i.e. 0x0d and 0x0a or `\r\n`).
- (Optional) Add one or more comments to the file.
- Write the global attribute Scheme, e.g.: `Scheme= "AKT-W15"`
- Write your global attributes, e.g.:

  ```
  Operator= "Mr. Miller"
  TC_Compressor= TA03-08F
  Barom= "1013 mbar"
  MinMax= 0  236.0
  ```

- Write the channel attributes `"Channelname"`, `"Unit"` and `"Datatype"`:

```
Channelname= $1
Unit= $2
Datatype= $3
```

- Write your own channel attributes.
- Write the keyword `UXX-END` followed by new line separator (i.e. 0x0d and 0x0a or `\r\n`).
- Write the channelnames (separated by a tab character):

```
Time        Speed    Torq     bsfc
hh:mm:ss    rpm      Nm       g/kWh
time        real8    real8    real8
```

- Write the channel data (separated by a tab character):

```
12:02:53 1000.2    32.23    267.6
12:03:47 1501.8    42.45    284.5
12:05:12 2004.2    48.44    296.3
```

8.3.14 Import of simple TEXT files using the UTX-Filter.

The data matrix for the simple Text-Import-Filter must be complete. If values are missing, for example are marked as ** the import will fail.

With the help of the UTX filter it is relativly simple to write an import filter with UniScript. Here is an UniScript example for the following text file:

```
Speed    Torque   Beewg   NOx
1/min    Nm       g/kWh   ppm
1000,2    32,23    267,6 990
1501,8    42,45    **    1100
2004,2    48,44    296,3 1200
```

UniScript:

```
ADDIN_AddToUI(ADDIN_FILE_IMPORT, "Test Cell File (*.ascii)|*.ascii|", ..
                                 "_My_Import");
// Check if the filename ends with .ascii
def IsFileMyImport_ASCII(ssFileName)
{
    if (strlower(SplitPath(ssFileName)[4]) == ".ascii") {
        return TRUE;
    }
    return FALSE;
}
// Returns the UTX header
def _My_GetHeader()
{
    ssHeader = "[[UXX-BEGIN
              Creator= Test Cell ASCII FILTER
              Channelname =  $1
              unit = $2
```

```
            UXX-END
          ]]";
    return ssHeader;
}
def _My_Import(ssFileName)
{
    if (IsFileMyImport_ASCII(ssFileName) == FALSE) {
        // File name extension is not .ascii
        return "";
    }
    if (IsFunctionLoaded("UXX_Import") == FALSE) {
        MessageBoxError("Cannot import file because the UTX-Filter is " + ..
                    "not loaded.\n" + ..
                    "Choose Tools=Addin-Manager an mark the Addin UTX-Filter.");
        return "#IMPORTERROR#";
    }
    // Writes the header to a temp file.
    // The temp file will be deleted at the end of the function.
    ssHeader = _My_GetHeader();
    ssTempFile = GetTempFileName();
    fp = fopen(ssTempFile, "wt");
    if (fp == 0) {
        return "#IMPORTERROR#";
    }
    fwrite(fp, "char", ssHeader);
    fclose(fp);
    // Call the UTX-Filter:
    ssRet = UXX_Import([ssFileName, ssTempFile]);
    // Delete temp file:
    DeleteFile(ssTempFile);
    return ssRet;
}
```

The example uses the UTX-Filter to import the test data. The UTX filter function UXX_Import()is invoked with two file names. The two file names are passed as a string vector. The first element is the file name of the data file and the second element is the file name of the header file.

In our example the header is specified in the _My_GetHeader() function. The header is saved in a temp file.

The header specifies the position of the channel names und units.

The program code can be saved in an ic file in the UniPlot-autoload directory. The file must have the extension `.ic`, for example `my_import.ic`. During the startup of UniPlot all ic files in the autoload directory will be loaded.

8.4 Import 3D Matrix Data

To import a matrix with z-values from a Text or Excel file:

- Choose *datamore-data-functions*.
- Select the **Load Matrix** function from the list box and then choose *OK*.

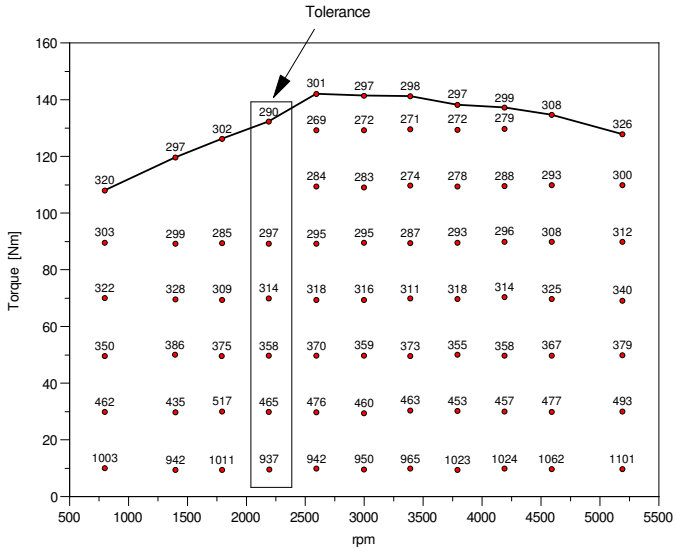
- In the **File Open** dialog box, select the type of data file. Select the name of the data file and then click the *OK* button.
- If the file contains a valid data matrix, a dialog box will be displayed specifying the coordinates of the corner points.

The number of data rows and columns must lie between 2 and 1000. The number of columns and rows do not have to be equal.

8.5 Definition of the Full Load Line (WOT)

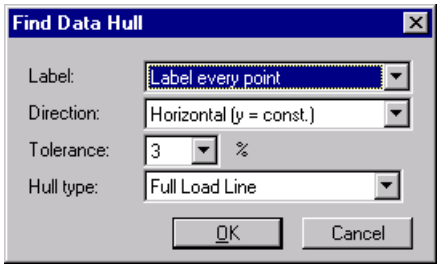
8.5.1 Finding the Data Hull

UniPlot offers a function to find the data hull of data points measured as map cross sections (as shown in the following diagram). The function finds the maximum and minimum points of each cross section band.



The x-values of one cross section must lie inside a tolerance band as shown in the diagram. The width of the tolerance band can be specified in the **Find Data Hull** dialog box. The maximum and minimum points of each cross section are connected to form the hull. A choice of 3 hulls is available.

To execute the function, select the dataset and choose *datamore-data-functions*. Choose **Find Data Hull (Full Load Line)** from the list to open the following dialog box:



Select the appropriate options and click OK to execute the function. The new hull will be inserted into the dataset and saved as a **User Hull**.

UniPlot computes a convex hull curve during the import of a 3D dataset. The Full Load Line is the "north" part of the data hull. This convex curve may exclude some of the Full Load Line points. To plot the line correctly, the data file must be edited by entering characters in the last column to mark the Full Load Line's data points. To edit a text (ASCII) data file, use an editor or spreadsheet program.

The following characters can be used to define a data hull:

A: Start point on the Full Load Line.

The point must belong to the automatically calculated "north" part of the convex hull and must have a smaller x-coordinate than the end point, **E**.

V: Full Load Line Point.

E: End point of the Full Load Line.

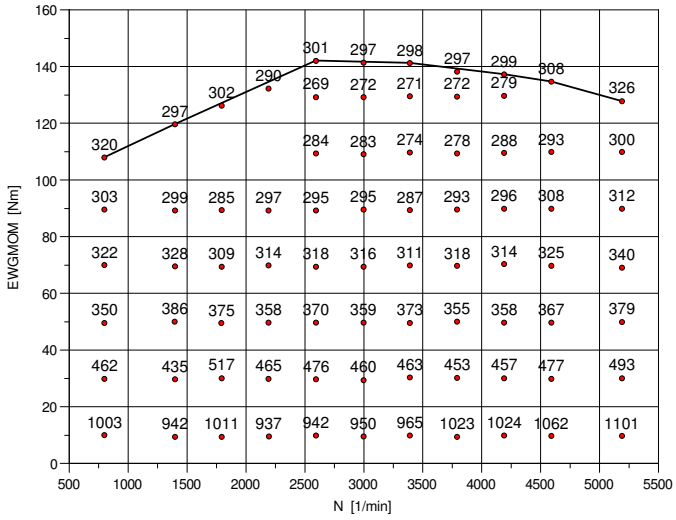
The point must belong to the automatically calculated "north" part of the convex hull and must have a greater x-coordinate than the start point, **A**.

If the symbols are entered as lower case letters (a, v, e), the corresponding data points of the Full Load Line (WOT) will not be labeled. However, use the lower case letters when you want to avoid label overlap. This can occur when the data points on the curve lie close together.

Procedure Hints

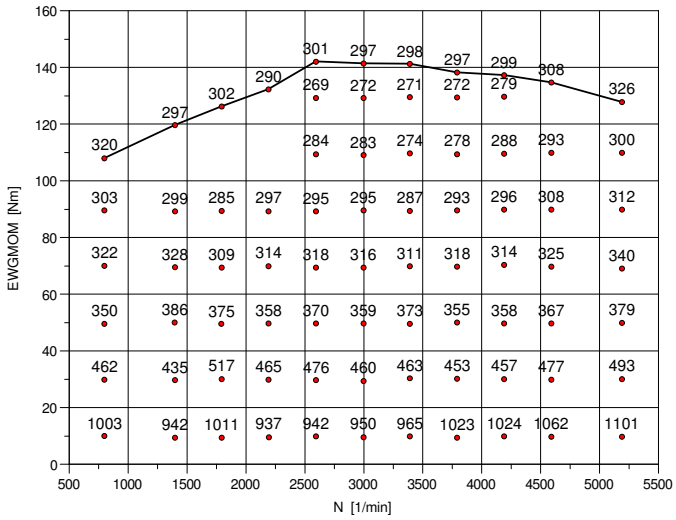
To create the Full Load Line:

- Import the 3D dataset into UniPlot.
- Open the 3D dataset dialog box, choose the **Data Point Symbols** dialog page and select the **Scatter Plot** and the **Show z-Values** check boxes. In the **Hull** dialog page, choose the **Entire Hull** hull type and then choose *OK*.
- Load the data file to a program in which you can edit. For a text file you can use a UniPlot editor (Choose the Open button from the toolbar and select the file name). This is what you will see:



As you see, the data points with the z-values 320, 297, 301, 298, 299, 308, 326 were computed as the Full Load Line.

- With the help of the scatter plot and the Full Load Line you can easily determine the true Full Load Line. Mark, as described above, the start point, end point and all other Full Load Line points. In the following figure, the data row with the z-value, 320, is marked as the start point (**A**). The data rows containing the 297, 302, 290, 301, 297, 298, 297, 299, 308 z-values are marked with a **V**. The row with the z-value 326 is marked as the end point of the Full Load Line (**E**).



Save the changes, close the data file and import the dataset again.

8.6 Arbitrary Data Hull

UniPlot makes it possible to load an arbitrary data hull, which can also contain islands, into a 3D dataset.

Only isolines within the data hull of the contour map will be plotted. Only surface facets of the 3D Plot lying completely inside the hull will be plotted.

The data hull must be a three columned ASCII File. The first and second column contain the x- and y-coordinates and the third column contains a control letter. Hull coordinates must be entered as world coordinates e.g. the same coordinate system as the 3D data points.

The following 4 letters must be used as control letters:

m m is the short form of Move To. The letter can be used to reach islands without inserting a connecting line.

M M has the same meaning as **m**, though the point will be labeled by the z-value of the contour map at this position.

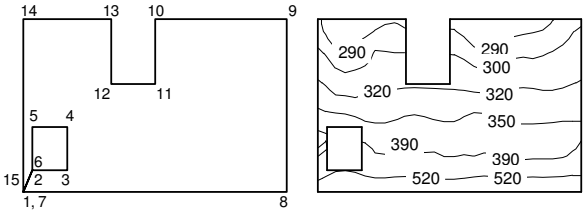
l l is the short for Line To. This letter ensures that a visible line is drawn.

L L has the same meaning as **l**, but the point will be labeled by the z-value of the contour map at this position.

Example:

In the following table is the hull's ASCII file. The numbers in brackets relate to the numbers in the figure. They should not be entered into the hull file.

```
1500    20  m   (1)
1600    30  m   (2)
2000    30  l   (3)
2000    50  l   (4)
1600    50  l   (5)
1600    30  l   (6)
1500    20  m   (7)
4500    20  l   (8)
4500    100 l   (9)
3000    100 l   (10)
3000    70  l   (11)
2500    70  l   (12)
2500    100 l   (13)
1500    100 l   (14)
1500    20  l   (15)
```



8.7 Save Hull

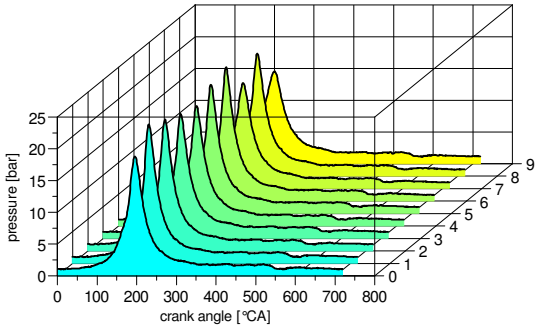
To save a 3D dataset hull, select the dataset and then choose *filemore-file-functions*. Select the **Save Hull** option and then click **OK**. In the **SaveAs** dialog box that appears, choose the name of the data file in which you want to save the hull and click **OK**.

8.8 Load Hull

To load a data hull into a 3D dataset, select the dataset and then choose *filemore-file-functions*. Select the **Load Hull** option and then choose **OK**. In the **File Open** dialog box that appears, choose the hull data file and click **OK**.

8.9 Waterfall Data Import

In a waterfall diagram, datasets are stacked into the depth of the diagram.



Data can be imported into a waterfall diagram in the same way as they are imported into a 2D diagram.

Two addional import data functions are available:

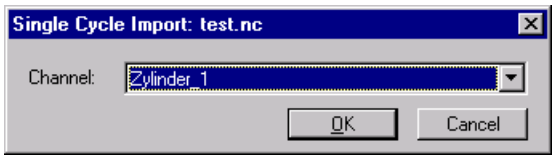
- Import of cycles from a single channel This function splits perodical data into cycles. (*filemore-file-functionswaterfall-single-cycle-import*)
- Import of multiple channels This function imports data from multiple channels. One channel can be defined as an x-axis channel. A dataset is created for each selected channel. (*filemore-file-functionswaterfall-multi-channel-import*).

8.9.1 Import of cycles from a single channel

This function splits perodical data into cycles.

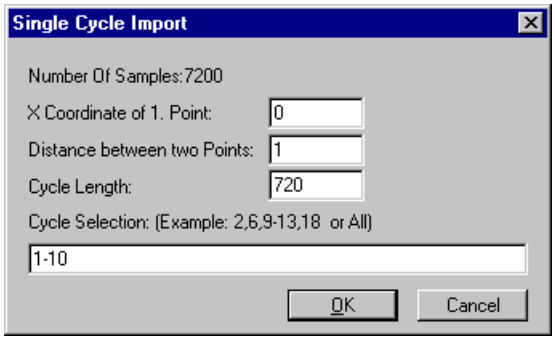
To import the data:

- Choose *filemore-file-functions*.
- Select the list item **Waterfall-Single Cycle Import**.
- Select a data file from the **File Open** dialog box.
- If the file was read successfully, a dialog box with the channel names will appear:



  Choose a channel and click the **OK** button.

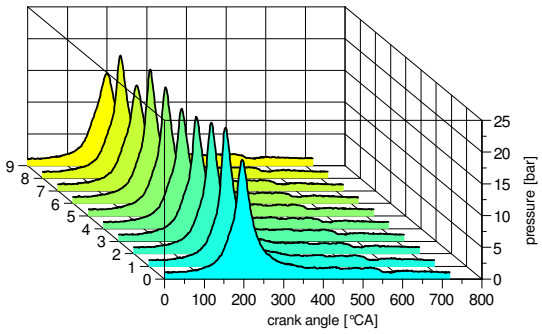
- In the following dialog box you can specify the cycle you would like to import:



In this example, the cycles 1 to 10 were imported with a **Cycle Length** of 720 degrees CA. A crank angle of 0 grad CA was set for the 1st data point and 1 degree CA as step width.

If the diagram is not switched to the waterfall diagram, click the **Waterfall** button. To scale the diagram axes, click the **Zoom Out** button.

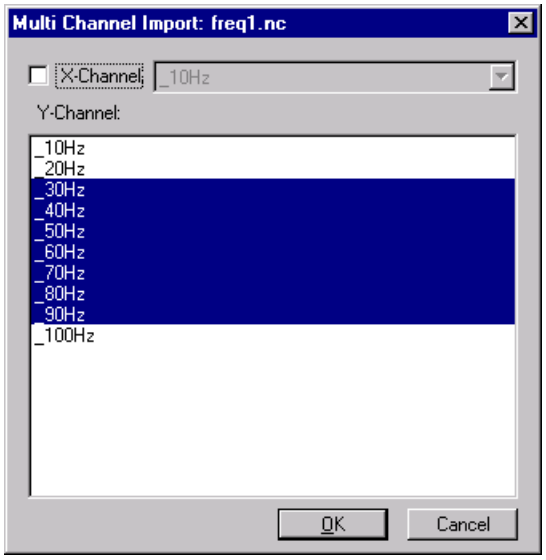
To change the orientation of the waterfall diagram, position the cursor in the yellow handle of the upper right corner and drag the corner to the desired position. Color filling of curves can be adjusted with the function **Data=>More Data Functions=>Waterfall Data Configuration**.



8.9.2 Import of multiple channels

This function imports data from multiple channels. One channel can be defined as an x-axis channel. A dataset is created for each selected channel.

To import multiple channels, choose *filemore-file-functionswaterfall-multi-channel-import*. The following dialog box will be displayed:



Select the channels to be imported. If necessary, you can specify one channel as a x-channel.

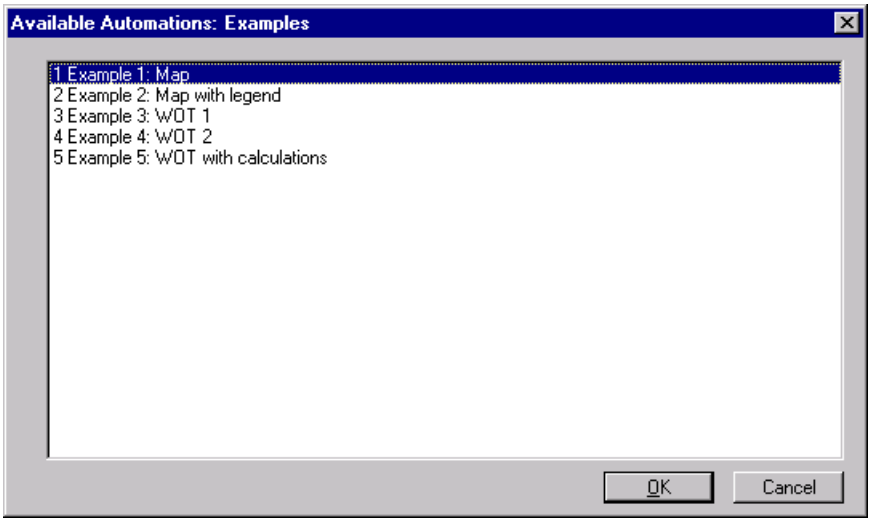
CHAPTER 9

Overview Automation

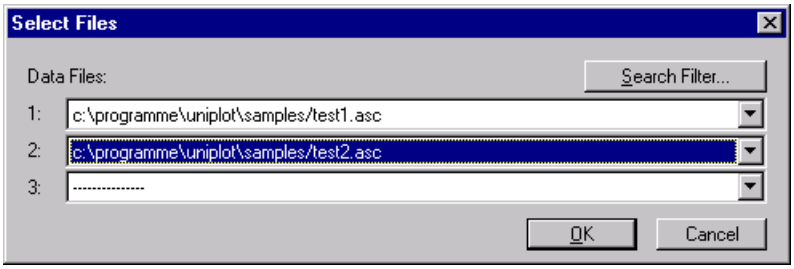
When creating a number of similar reports, it is recommended to automate this task.

9.1 Automation from the user's point of view

Choose a category from the **Report** menu, e. g. Map Plots. A list of available reports will be displayed. Example:

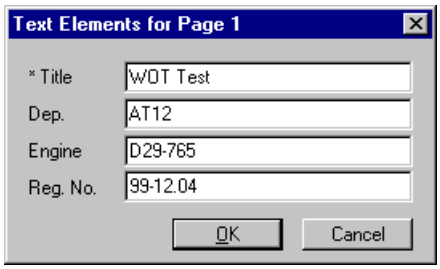


Select a report. The next step is to select data files from the following dialog:



Up to 16 data files can be selected depending on the chosen automation script.

A template will be opened and the data will be loaded into the diagrams. Header data will be inserted into the template. The creation of the page ends when a dialog box for the input of additional text data appears:



9.2 Creating an automation

An automation consists of two files:

- **IC file**: The ic file (script file) contains the program which will control the execution of the automation. The script file should have the file name extension `.ic`.
- **IPW file**: The ipw file (UniPlot document) contains the template with diagrams, text fields logos etc. The UniPlot document should have the file name extension `.ipw`.

The ic and the ipw files should be saved in the same directory. To install the automation, the directory where the two files are stored must be specified in the *tools-more-options* dialog box, in the field **Search path for automation script files** (e.g. `c:\uniplot\samples\automate`). Multiple directories can be specified and must be separated by a semi-colon (;). After the dialog box is closed, all IC files in the specified directories will be loaded.

9.2.1 The template file (.ipw)

The template is a ordinary UniPlot document containing diagrams, text and drawing objects, logos, OLE objects and all other possible UniPlot objects.

9.2.2 Diagrams

The scaling of the diagram axes should be set up in the template in such a way that the diagrams will display the interesting part of the data. If the data varies significantly, the axes can be autoscaled in the script file. x- and y-axes can be scaled independently (see *auto_ScaleAxes*). Each diagram on a page has an unique name. The name will be used in the script file to access a diagram and to add data to a diagram.

9.2.3 Text Objects

The template can contain any number of text place-holders which will be replaced during the execution of an automation script by user input or by text received from the data file. The place-holder is text that begins with a dollar sign, followed by the text, followed by another dollar sign. Example: `$Engine$`.

To set an order to display the place-holder text in the dialog box, an optional number can be added to the place-holder. Example: `$(1)Title$`. The number must be surrounded by brackets and must be placed behind the first dollar sign.

If the text of a place-holder should be received from a data file, the spelling and case must be identical to one global attribute in the NC file. Normally, every NC file created by UniPlot has a global attribute `"Origin"` containing the file name of the original data file. The placeholder text for this attribute will be `$Origin$`. To read the attribute value of a variable, the variable name must be followed by a period followed by the name of the attribute. Example: `$Torque.long_name$`.

9.2.4 Legends and Field Functions

Every page can have one legend. The legend is a normal text object. (To create a text object, select the A button from the toolbar and drag the mouse). The text of the text object must appear in this format: `$LEGEND$`.

The template can contain field functions. Multiple legends can be created (see *Field Functions*) for a page by using field functions. To create a legend for a specific diagram, add a text object to the diagram and type in the text `@f{legend}`.

To create a field function that displays the document name, create a text element and type `@f{documentname}` into the element (*filemore-file-functions* item **Add filename to document page**).

A template file can contain multiple pages. Which page is loaded and in what order will be specified in the ic-file.

9.2.5 The ic file

The script file (.ic) is a text file containing a program which controls the execution of the automation. To create a script file, open a program editor (*file-new*). To edit an existing script file, choose *file-open*.

The following examples are used to describe different sections of an automation script file. The examples can be found in the directory `UniPlot\Samples\Automate`.

To enable UniPlot to execute the following examples, load the automation scripts into UniPlot. Choose *tools-more-options*. In the dialog box that appears, specify the path of the ic file in the **Search Path for Automation Script Files** text field (e.g. `C:/Program Files/Uniplot/Samples/Automate/`). Click the *OK* button. A new **Reports** menu should be added to the main menu.

Select the category **Examples=>Example 1: Map**. In the data file dialog box that appears, select the example file `map-data.xls`.

```
auto_AddToUI("Examples", "Example 1: Map", "RS_Example1");

def RS_Example1()
{
    MessageBox("In the following dialog please select the file map-data.xls");
    auto_SetFileNameDialogInit("*.xls", GetRootDirectory() + "samples/");
    svFile = auto_GetFileNameDialog(1);
    if (svFile[1] == "DLG_CANCEL") {
        return;
```

```
    }
    auto_LoadTemplate(GetRootDirectory() + "samples/automate/Example.ipw");
    auto_ImportData(svFile[1]);
    auto_LoadDataset("Diagramm 1", "N", "EWGMOM", "EWGLST");
    auto_LoadDataset("Diagramm 2", "N", "EWGMOM", "BEEWG");
    auto_ScaleAxes();
    auto_UpdatePage();
}
```

The function **RS_Example1** controls the execution of the automation. This function is called when the user selects the respective menu **Report=>Example 1: Map**.

The function call *auto_AddToUI* (UI for User-Interface) inserts a new menu item to the Reports menu, if this menu item is not already available, and adds a new item to the listbox (`"Example 1: Map"`).

The script function **RS_Example1** will then be registered so that **Reports=>Examples=>Example 1: Map** can be called from the menu.

The function has three parameters:

auto_AddToUI

For each automation the *auto_AddToUI* function will be called up once.

The first parameter sets the name of the menu to which the evaluation should be assigned.

The function's second parameter (`"Example 1: Map"`) will appear in the dialog box when the **Evaluation=>Example-Evaluation** menu is chosen.

The third parameter (`"RS_Example1"`) is the name of the function to be called up. The name can be chosen freely. Because each function name in UniScript can only be used once, make sure the name does not already exist. It is recommended to start the name with a prefix not already in use by UniPlot e.g. `rs_` or `abc_` etc.

With the help of the *what* function, it is possible to check if functions with the prefix already exist in UniPlot. To do this, open the command window and enter, for example, `what("RSB_*")`. A list will appear with all the functions that begin with `RSB_`. The script file should be saved with a template file in the same directory. For example: `uniplot\template\auto`.

Automations can be located in various directories. The search directory is set in the *tools-more-options* dialog box. Search directories are entered in the **Search Path for automation script files**. The paths must be separated by a semi-colon (;). Only automation script files should be saved in these directories.

When starting UniPlot, the script files will be searched and opened in the paths given. The script files must have the name extension .ic.

All UniScript functions can be called up within a script function. To simplify the writing of script files, prepared functions are available for standard functions such as Choose Files, Search and Replace Text, and for the calculation of data.

These functions are briefly described here:

auto_GetFileNameDialog

The user can choose data files with this function. The function returns the name of the chosen file as a vector. If the **Cancel** button is chosen, the character string `"DLG_CANCEL"` will be returned as the first element of

the vector. If more than 4 files should be chosen, the number is entered in the function as the first parameter. For example, calling `auto_GetFileNameDialog(8)` gives the user the possibility to choose up to 8 files. A maximum of 16 files can be chosen.

auto_ReplaceTextDialog

This function searches the template page for text matching the place holder pattern (`$Text$`). If the text is found in the template page, a dialog box will appear. With its help, the text can be entered. If a NC file name (return value of the *auto_ImportData* function) is entered, the function tries to find the place-holder text in the global attributes or the variable attributes of the NC file. All of the loaded text from the NC file will be marked with an astrix in the dialog box.

auto_LoadTemplate

This function opens a template and creates a copy of the required page. The copied page will be inserted into either a new document or in the active document.

The first parameter is the name of the template. If the template originates from one of the entered automation script directories, the path should not be entered.

The second parameter is the name of the page. If the page is to be inserted to the active document, the *auto_AddPage* function should be called first. (See Example **RS_Example4**).

The function returns the handle of the created page. The handle can be used, for example, to retrieve the handle of a diagram (*PageGetLayerHandle*).

auto_ImportData

This function opens the given data file and converts it to NC format. The file name is given as the first parameter. The name of the NC file will be returned as a return value. The NC file name can be used, for example, for the *auto_ReplaceTextDialog* function.

The name of the created NC file will be saved as a global variable. The *auto_LoadDataset* function uses this variable to load data from the NC file to a given diagram.

auto_LoadDataset

The *auto_LoadDataset* function creates an 1D, 2D or 3D dataset. The dataset will be inserted into the given diagram. The function returns a handle which will be used to access the dataset.

The dataset style will be taken from the active style file. The first dataset in the diagram receives the first style from the file, the second dataset receives the second style in the file and so on.

To set the style for the next dataset being created by *auto_LoadDataset*, call the *auto_SetDatasetStyle* function. The style can be passed by name or index.

auto_UpdatePage

This function should be called in order to finish the creation of a page.

Create a map with a color legend:

```
auto_AddToUI("Examples", "Example 2: Map with legend", "RS_Example2");

def RS_Example2()
{
    MessageBox("In the following dialog please select the file map-data.xls");
```

```
    auto_SetFileNameDialogInit("*.xls", GetRootDirectory() + "samples/");
    svFile = auto_GetFileNameDialog(1);
    if (svFile[1] == "DLG_CANCEL") {
        return;
    }
    auto_LoadStyleFile(GetRootDirectory() + "samples/automate/style1.icb");
    hPage = auto_LoadTemplate(GetRootDirectory() + "samples/automate/Example.ipw");
    auto_ImportData(svFile[1]);
    hData = auto_LoadDataset("Diagramm 1", "N", "EWGMOM", "EWGLST");
    auto_xyz_CreateColorLegend(hData)
    auto_LoadDataset("Diagramm 2", "N", "EWGMOM", "BEEWG");
    auto_ReplaceText("$X-Title$", "Dies ist die Y-Achse");
    auto_ReplaceTextDialog()
    auto_UpdatePage();
}
```

In the following example, up to 3 data files can be chosen:

```
auto_AddToUI("Examples", "Example 3: WOT 1", "RS_Example3");

def RS_Example3()
{
    MessageBox("In the following dialog please select the file VOLLAST.ASC");
    auto_SetFileNameDialogInit("*.asc; *.nc", GetRootDirectory() + "samples/");
    svFile = auto_GetFileNameDialog(3);
    if (svFile[1] == "DLG_CANCEL") {
        return;
    }
    NumberOfFiles = len(svFile);

    hPage = auto_LoadTemplate("example");

    for (i in 1:NumberOfFiles) {
        svFile[i] = auto_ImportData(svFile[i]);
        auto_LoadDataset("Diagramm 1", "N", "MEFF");
        auto_LoadDataset("Diagramm 2", "N", "EWGLST");
    }
    auto_ScaleAxes();
    auto_ReplaceTextDialog(svFile[1]);
    auto_UpdatePage();
}
```

The following example creates a document with 2 pages. The first page shows the data (N, MEFF) and (N, EWGLST) and the second page displays the data (N, BEFF) and (N, TOEL1).

```
auto_AddToUI("Examples", "Example 4: WOT 2", "RS_Example4");

def RS_Example4()
{
    MessageBox("In the following dialog please select the file VOLLAST.ASC");
    auto_SetFileNameDialogInit("*.asc; *.nc", GetRootDirectory() + "samples/");
    svFile = auto_GetFileNameDialog(3);
    if (svFile[1] == "DLG_CANCEL") {
```

```
        return;
    }
    NumberOfFiles = len(svFile);

    hPage = auto_LoadTemplate("example");

    for (i in 1:NumberOfFiles) {
        svFile[i] = auto_ImportData(svFile[i]);
        auto_LoadDataset("Diagramm 1", "N", "MEFF");
        auto_LoadDataset("Diagramm 2", "N", "EWGLST");
    }
    auto_ScaleAxes();
    auto_ReplaceTextDialog(hPage, svFile[1]);
    auto_UpdatePage();

    // Add the next page to the active Page
    auto_AddPage();

    hPage = auto_LoadTemplate("example");

    for (i in 1:NumberOfFiles) {
        svFile[i] = auto_ImportData(svFile[i]);
        auto_LoadDataset("Diagramm 1", "N", "BEFF");
        auto_LoadDataset("Diagramm 2", "N", "TOEL1");
    }
    auto_ScaleAxes();
    auto_ReplaceTextDialog(hPage, svFile[1]);
    auto_UpdatePage();
}
```

This example shows how the average value of two curves can be calculated and how two datasets can be divided.

```
auto_AddToUI("Examples", "Example 5: WOT with calculations", "RS_Example5");

def RS_Example5()
{
    MessageBox("In the following dialog please select the file VOLLAST.ASC");
    auto_SetFileNameDialogInit("*.asc; *.nc", GetRootDirectory() + "samples/");
    svFile = auto_GetFileNameDialog(2);
    if (svFile[1] == "DLG_CANCEL") {
        return;
    }
    NumberOfFiles = len(svFile);
    if (NumberOfFiles != 2) {
        MessageBox("Es müssen 2 Dateien ausgewählt werden");
        return FALSE;
    }

    hPage = auto_LoadTemplate("example");

    svFile[1] = auto_ImportData(svFile[1]);
    hData1 = auto_LoadDataset("Diagramm 1", "N", "MEFF");
```

```
    svFile[2] = auto_ImportData(svFile[2]);
    hData2 = auto_LoadDataset("Diagramm 1", "N", "MEFF");

    // Retrieve the handel of Diagramm 2
    hLayer2 = PageGetLayerHandle(hPage, "Diagramm 2");

    // Calculate Mean dataset
    auto_xy_Mean(hData1, hData2);

    // Calculate Div dataset and add to "Diagramm 2".
    auto_xy_Div([hData1, hLayer2], hData2);

    // Autoscale axes
    auto_ScaleAxes();
    auto_ReplaceTextDialog(hPage, svFile[1]);
    auto_UpdatePage();
}
```

The function *auto_xy_Mean* calculates the average of 2 datasets (curves). The datasets must have ascending x-coordinates and overlap in the x-direction. The x-coordinates of the two datasets do not have to be identical.

Datasets will be passed to the function with their handles. Handles are returned from the function *auto_LoadDataset*.

The *auto_xy_Mean* function, like the *auto_LoadDataset* function, returns the handle of the created dataset. If the dataset cannot be created, the function returns the value 0.

To delete a dataset from a diagram, call up the *XYDestroy* or *XYZDestroy* function.

CHAPTER 10

Licenses

10.1 UniPlot

```
Grant of License: This agreement grants the buyer a
non-exclusive and transferable license to use UniPlot for Windows.

Usage: The buyer is granted the right to use UniPlot for
Windows on a single computer. The buyer is allowed to make one copy of
the program solely for backup or archival purposes. The buyer may not
rent or lease the program but may transfer their rights under this
agreement on a permanent basis if all copies of the program and all
written materials are transferred and the third party agrees to the
terms of this agreement.

Warranty: (1.) There is a 6 month warranty for errors
discovered in the program. An error is recognized as such when the
program function does not agree with the function description in the
manual. (2.) No liability will be accepted for any loss, damage or
injury occurring from the supply or use of the program. In every
instance, liability is limited to the amount actually paid for the
program.

Other: (1.) Changes or additions to this agreement can be
arranged only in writing. (2.) If a single condition of this agreement
is found to be improper, the remaining conditions of the license
agreement will not be affected. (3.) Legal domicile of this agreement
is Berlin, Germany.

Uniplot Software GmbH,  D-15834 Rangsdorf, Germany
```

10.2 netCDF

```
Copyright 1993-2002 University Corporation for Atmospheric Research/Unidata

Portions of this software were developed by the Unidata Program at the University
Corporation for Atmospheric Research.
```

```
Access and use of this software shall impose the following obligations and
understandings on the user. The user is granted the right, without any fee or
cost, to use, copy, modify, alter, enhance and distribute this software, and any
derivative works thereof, and its supporting documentation for any purpose
whatsoever, provided that this entire notice appears in all copies of the
software, derivative works and supporting documentation. Further, UCAR requests
that the user credit UCAR/Unidata in any publications that result from the use of
this software or in any product that includes this software. The names UCAR
and/or Unidata, however, may not be used in any advertising or publicity to
endorse or promote any products or commercial entity unless specific written
permission is obtained from UCAR/Unidata. The user also understands that
UCAR/Unidata is not obligated to provide the user with any support, consulting,
training or assistance of any kind with regard to the use, operation and
performance of this software nor to provide the user with any updates, revisions,
new versions or "bug fixes."

THIS SOFTWARE IS PROVIDED BY UCAR/UNIDATA "AS IS" AND ANY EXPRESS OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT
SHALL UCAR/UNIDATA BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES
OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN
AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN
CONNECTION WITH THE ACCESS, USE OR PERFORMANCE OF THIS SOFTWARE.
```

10.3 Scintilla

```
License for Scintilla and SciTE

Copyright 1998-2002 by Neil Hodgson <neilh@scintilla.org>

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its
documentation for any purpose and without fee is hereby granted,
provided that the above copyright notice appear in all copies and that
both that copyright notice and this permission notice appear in
supporting documentation.

NEIL HODGSON DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS
SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY
AND FITNESS, IN NO EVENT SHALL NEIL HODGSON BE LIABLE FOR ANY
SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,
WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER
TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE
OR PERFORMANCE OF THIS SOFTWARE.
```

10.4 VTK

Copyright (c) 1993-2001 Ken Martin, Will Schroeder, Bill Lorensen All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither name of Ken Martin, Will Schroeder, or Bill Lorensen nor the names of any contributors may be used to endorse or promote products derived from this software without specific prior written permission.

* Modified source versions must be plainly marked as such, and must not be misrepresented as being the original software.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

10.5 PCRE

PCRE LICENCE

PCRE is a library of functions to support regular expressions whose syntax and semantics are as close as possible to those of the Perl 5 language.

Release 8 of PCRE is distributed under the terms of the "BSD" licence, as specified below. The documentation for PCRE, supplied in the "doc" directory, is distributed under the same terms as the software itself.

The basic library functions are written in C and are freestanding. Also included in the distribution is a set of C++ wrapper functions.

THE BASIC LIBRARY FUNCTIONS

```
Written by:       Philip Hazel
Email local part: ph10
Email domain:     cam.ac.uk

University of Cambridge Computing Service,
Cambridge, England.

Copyright (c) 1997-2009 University of Cambridge
All rights reserved.

THE C++ WRAPPER FUNCTIONS
-------------------------

Contributed by:   Google Inc.

Copyright (c) 2007-2008, Google Inc.
All rights reserved.

THE "BSD" LICENCE
-----------------

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

        * Redistributions of source code must retain the above copyright notice,
          this list of conditions and the following disclaimer.

        * Redistributions in binary form must reproduce the above copyright
          notice, this list of conditions and the following disclaimer in the
          documentation and/or other materials provided with the distribution.

        * Neither the name of the University of Cambridge nor the name of Google
          Inc. nor the names of their contributors may be used to endorse or
          promote products derived from this software without specific prior
          written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.

End
```

10.6 zlib

```
(C) 1995-2002 Jean-loup Gailly and Mark Adler

 This software is provided 'as-is', without any express or implied
 warranty.  In no event will the authors be held liable for any damages
 arising from the use of this software.

 Permission is granted to anyone to use this software for any purpose,
 including commercial applications, and to alter it and redistribute it
 freely, subject to the following restrictions:

 1. The origin of this software must not be misrepresented; you must not
    claim that you wrote the original software. If you use this software
    in a product, an acknowledgment in the product documentation would be
    appreciated but is not required.
 2. Altered source versions must be plainly marked as such, and must not be
    misrepresented as being the original software.
 3. This notice may not be removed or altered from any source distribution.

 Jean-loup Gailly        Mark Adler
 jloup@gzip.org          madler@alumni.caltech.edu
```

10.7 LZF

```
Copyright (c) 2000-2007 Marc Alexander Lehmann <schmorp@schmorp.de>

Redistribution and use in source and binary forms, with or without modifica-
tion, are permitted provided that the following conditions are met:

  1.  Redistributions of source code must retain the above copyright notice,
      this list of conditions and the following disclaimer.

  2.  Redistributions in binary form must reproduce the above copyright
      notice, this list of conditions and the following disclaimer in the
      documentation and/or other materials provided with the distribution.


THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MER-
CHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.  IN NO
EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPE-
CIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS;
OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTH-
ERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
OF THE POSSIBILITY OF SUCH DAMAGE.
```

10.8 expat

```
Copyright (c) 1998, 1999, 2000 Thai Open Source Software Center Ltd
                               and Clark Cooper
Copyright (c) 2001, 2002 Expat maintainers.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

10.9 qhull

```
                    Qhull, Copyright (c) 1993-2002

        The National Science and Technology Research Center for
         Computation and Visualization of Geometric Structures
                           (The Geometry Center)
                         University of Minnesota
                              400 Lind Hall
                           207 Church Street S.E.
                        Minneapolis, MN 55455  USA

                         email: qhull@geom.umn.edu

This software includes Qhull from The Geometry Center.  Qhull is
copyrighted as noted above.  Qhull is free software and may be obtained
via http from www.geom.umn.edu.  It may be freely copied, modified,
and redistributed under the following conditions:

1. All copyright notices must remain intact in all files.

2. A copy of this text file must be distributed along with any copies
   of Qhull that you redistribute; this includes copies that you have
   modified, or copies of programs or other software products that
   include Qhull.
```

```
3. If you modify Qhull, you must include a notice giving the
   name of the person performing the modification, the date of
   modification, and the reason for such modification.

4. When distributing modified versions of Qhull, or other software
   products that include Qhull, you must provide notice that the original
   source code may be obtained as noted above.

5. There is no warranty or other guarantee of fitness for Qhull, it is
   provided solely "as is".  Bug reports or fixes may be sent to
   qhull_bug@geom.umn.edu; the authors may or may not act on them as
   they desire.
```

10.10 jpeg

```
This software is copyright (C) 1991-1998, Thomas G. Lane.
All Rights Reserved except as specified below.

Permission is hereby granted to use, copy, modify, and distribute this
software (or portions thereof) for any purpose, without fee, subject to these
conditions:
(1) If any part of the source code for this software is distributed, then this
README file must be included, with this copyright and no-warranty notice
unaltered; and any additions, deletions, or changes to the original files
must be clearly indicated in accompanying documentation.
(2) If only executable code is distributed, then the accompanying
documentation must state that "this software is based in part on the work of
the Independent JPEG Group".
(3) Permission for use of this software is granted only if the user accepts
full responsibility for any undesirable consequences; the authors accept
NO LIABILITY for damages of any kind.
```

10.11 png

```
This copy of the libpng notices is provided for your convenience.  In case of
any discrepancy between this copy and the notices in the file png.h that is
included in the libpng distribution, the latter shall prevail.

COPYRIGHT NOTICE, DISCLAIMER, and LICENSE:

If you modify libpng you may insert additional notices immediately following
this sentence.

libpng versions 1.0.7, July 1, 2000, through 1.2.4, July 8, 2002, are
Copyright (c) 2000-2002 Glenn Randers-Pehrson
and are distributed according to the same disclaimer and license as libpng-1.0.6
with the following individuals added to the list of Contributing Authors
```

```
   Simon-Pierre Cadieux
   Eric S. Raymond
   Gilles Vollant

and with the following additions to the disclaimer:

   There is no warranty against interference with your enjoyment of the
   library or against infringement.  There is no warranty that our
   efforts or the library will fulfill any of your particular purposes
   or needs.  This library is provided with all faults, and the entire
   risk of satisfactory quality, performance, accuracy, and effort is with
   the user.

libpng versions 0.97, January 1998, through 1.0.6, March 20, 2000, are
Copyright (c) 1998, 1999 Glenn Randers-Pehrson, and are
distributed according to the same disclaimer and license as libpng-0.96,
with the following individuals added to the list of Contributing Authors:

   Tom Lane
   Glenn Randers-Pehrson
   Willem van Schaik

libpng versions 0.89, June 1996, through 0.96, May 1997, are
Copyright (c) 1996, 1997 Andreas Dilger
Distributed according to the same disclaimer and license as libpng-0.88,
with the following individuals added to the list of Contributing Authors:

   John Bowler
   Kevin Bracey
   Sam Bushell
   Magnus Holmgren
   Greg Roelofs
   Tom Tanner

libpng versions 0.5, May 1995, through 0.88, January 1996, are
Copyright (c) 1995, 1996 Guy Eric Schalnat, Group 42, Inc.

For the purposes of this copyright and license, "Contributing Authors"
is defined as the following set of individuals:

   Andreas Dilger
   Dave Martindale
   Guy Eric Schalnat
   Paul Schmidt
   Tim Wegner

The PNG Reference Library is supplied "AS IS".  The Contributing Authors
and Group 42, Inc. disclaim all warranties, expressed or implied,
including, without limitation, the warranties of merchantability and of
fitness for any purpose.  The Contributing Authors and Group 42, Inc.
assume no liability for direct, indirect, incidental, special, exemplary,
or consequential damages, which may result from the use of the PNG
Reference Library, even if advised of the possibility of such damage.
```

```
Permission is hereby granted to use, copy, modify, and distribute this
source code, or portions hereof, for any purpose, without fee, subject
to the following restrictions:

1. The origin of this source code must not be misrepresented.

2. Altered versions must be plainly marked as such and must not
   be misrepresented as being the original source.

3. This Copyright notice may not be removed or altered from any
   source or altered source distribution.

The Contributing Authors and Group 42, Inc. specifically permit, without
fee, and encourage the use of this source code as a component to
supporting the PNG file format in commercial products.  If you use this
source code in a product, acknowledgment is not required but would be
appreciated.


A "png_get_copyright" function is available, for convenient use in "about"
boxes and the like:

   printf("%s",png_get_copyright(NULL));

Also, the PNG logo (in PNG format, of course) is supplied in the
files "pngbar.png" and "pngbar.jpg (88x31) and "pngnow.png" (98x31).

Libpng is OSI Certified Open Source Software.  OSI Certified Open Source is a
certification mark of the Open Source Initiative.

Glenn Randers-Pehrson
randeg@alum.rpi.edu
July 8, 2002
```

10.12 cxImage

```
License

The class CxImage is free; as for the TIFF, JPEG, PNG and ZLIB
libraries : "If you use this source code in a product, acknowledgment
is not required but would be appreciated."
```

10.13 nullsoft Installer

```
license

Copyright (C) 1999-2001 Nullsoft, Inc.
```

```
This software is provided 'as-is', without any express or implied
warranty.  In no event will the authors be held liable for any damages
arising from the use of this software.

Permission is granted to anyone to use this software for any purpose,
including commercial applications, and to alter it and redistribute it
freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not
   claim that you wrote the original software. If you use this software
   in a product, an acknowledgment in the product documentation would be
   appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be
   misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.
```

10.14 libffi

```
license
libffi - Copyright (c) 1996-2008  Red Hat, Inc and others.
See source files for details.

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
``Software''), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED ``AS IS'', WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

10.15 HDF5

```
Copyright Notice and License Terms for
HDF5 (Hierarchical Data Format 5) Software Library and Utilities
-----------------------------------------------------------------------------

HDF5 (Hierarchical Data Format 5) Software Library and Utilities
Copyright 2006-2012 by The HDF Group.
```

NCSA HDF5 (Hierarchical Data Format 5) Software Library and Utilities
Copyright 1998-2006 by the Board of Trustees of the University of Illinois.

All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted for any purpose (including commercial purposes)
provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice,
   this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice,
   this list of conditions, and the following disclaimer in the documentation
   and/or materials provided with the distribution.

3. In addition, redistributions of modified forms of the source or binary
   code must carry prominent notices stating that the original code was
   changed and the date of the change.

4. All publications or advertising materials mentioning features or use of
   this software are asked, but not required, to acknowledge that it was
   developed by The HDF Group and by the National Center for Supercomputing
   Applications at the University of Illinois at Urbana-Champaign and
   credit the contributors.

5. Neither the name of The HDF Group, the name of the University, nor the
   name of any Contributor may be used to endorse or promote products derived
   from this software without specific prior written permission from
   The HDF Group, the University, or the Contributor, respectively.

DISCLAIMER:
THIS SOFTWARE IS PROVIDED BY THE HDF GROUP AND THE CONTRIBUTORS
"AS IS" WITH NO WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED. In no
event shall The HDF Group or the Contributors be liable for any damages
suffered by the users arising out of the use of this software, even if
advised of the possibility of such damage.

Contributors: National Center for Supercomputing Applications (NCSA) at
the University of Illinois, Fortner Software, Unidata Program Center (netCDF),
The Independent JPEG Group (JPEG), Jean-loup Gailly and Mark Adler (gzip),
and Digital Equipment Corporation (DEC).

10.16 kazlib

```
/* Copyright 1996-2012
 * Kaz Kylheku <kaz@kylheku.com>
 * Vancouver, Canada
 * All rights reserved.
 *
 * BSD License:
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 *   1. Redistributions of source code must retain the above copyright
 *      notice, this list of conditions and the following disclaimer.
 *   2. Redistributions in binary form must reproduce the above copyright
 *      notice, this list of conditions and the following disclaimer in
 *      the documentation and/or other materials provided with the
 *      distribution.
 *   3. The name of the author may not be used to endorse or promote
 *      products derived from this software without specific prior
 *      written permission.
 *
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
 * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.
 */
```

10.17 upfirdn (Polyphase FIR Resampling)

```
Copyright (c) 2009, Motorola, Inc

All Rights Reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

* Redistributions of source code must retain the above copyright notice,
this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

* Neither the name of Motorola nor the names of its contributors may be
used to endorse or promote products derived from this software without
specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
```

```
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

10.18 LZ4

```
LZ4 - Fast LZ compression algorithm
Copyright (C) 2011-2013, Yann Collet.
BSD 2-Clause License (http://www.opensource.org/licenses/bsd-license.php)

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

    * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
    * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You can contact the author at :
- LZ4 homepage : http://fastcompression.blogspot.com/p/lz4.html
- LZ4 source repository : http://code.google.com/p/lz4/
```

10.19 libxml2

```
Except where otherwise noted in the source code (e.g. the files hash.c,
list.c and the trio files, which are covered by a similar licence but
with different Copyright notices) all the files are:
```

Copyright (C) 1998-2003 Daniel Veillard. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is fur-
nished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FIT-
NESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
DANIEL VEILLARD BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CON-
NECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of Daniel Veillard shall not
be used in advertising or otherwise to promote the sale, use or other deal-
ings in this Software without prior written authorization from him.

10.20 Color Brewer

Apache-Style Software License for ColorBrewer software and ColorBrewer Color
Schemes

Copyright (c) 2002 Cynthia Brewer, Mark Harrower, and The Pennsylvania State
University.

Licensed under the Apache License, Version 2.0 (the "License"); you may not
use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed
under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR
CONDITIONS OF ANY KIND, either express or implied. See the License for the
specific language governing permissions and limitations under the License.

10.21 cURL Library (http://curl.haxx.se)

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2014, Daniel Stenberg, daniel@haxx.se.

```
All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose
with or without fee is hereby granted, provided that the above copyright
notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE
OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not
be used in advertising or otherwise to promote the sale, use or other dealings
in this Software without prior written authorization of the copyright holder.
```

10.22 OpenSSL

```
LICENSE ISSUES
==============

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of
the OpenSSL License and the original SSLeay license apply to the toolkit.
See below for the actual license texts. Actually both licenses are BSD-style
Open Source licenses. In case of any license issues related to OpenSSL
please contact openssl-core@openssl.org.

OpenSSL License
---------------

/* ====================================================================
 * Copyright (c) 1998-2011 The OpenSSL Project.  All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 *
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in
 *    the documentation and/or other materials provided with the
 *    distribution.
 *
 * 3. All advertising materials mentioning features or use of this
 *    software must display the following acknowledgment:
 *    "This product includes software developed by the OpenSSL Project
 *    for use in the OpenSSL Toolkit. (http://www.openssl.org/)"
```

```
 *
 * 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to
 *    endorse or promote products derived from this software without
 *    prior written permission. For written permission, please contact
 *    openssl-core@openssl.org.
 *
 * 5. Products derived from this software may not be called "OpenSSL"
 *    nor may "OpenSSL" appear in their names without prior written
 *    permission of the OpenSSL Project.
 *
 * 6. Redistributions of any form whatsoever must retain the following
 *    acknowledgment:
 *    "This product includes software developed by the OpenSSL Project
 *    for use in the OpenSSL Toolkit (http://www.openssl.org/)"
 *
 * THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY
 * EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE OpenSSL PROJECT OR
 * ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
 * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
 * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED
 * OF THE POSSIBILITY OF SUCH DAMAGE.
 * ====================================================================
 *
 * This product includes cryptographic software written by Eric Young
 * (eay@cryptsoft.com).  This product includes software written by Tim
 * Hudson (tjh@cryptsoft.com).
 *
 */

Original SSLeay License
-----------------------

/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *
 * This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to.  The following conditions
 * apply to all code found in this distribution, be it the RC4, RSA,
 * lhash, DES, etc., code; not just the SSL code.  The SSL documentation
 * included with this distribution is covered by the same copyright terms
 * except that the holder is Tim Hudson (tjh@cryptsoft.com).
 *
 * Copyright remains Eric Young's, and as such any Copyright notices in
```

```
 * the code are not to be removed.
 * If this package is used in a product, Eric Young should be given attribution
 * as the author of the parts of the library used.
 * This can be in the form of a textual message at program startup or
 * in documentation (online or textual) provided with the package.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 *    must display the following acknowledgement:
 *    "This product includes cryptographic software written by
 *     Eric Young (eay@cryptsoft.com)"
 *    The word 'cryptographic' can be left out if the rouines from the library
 *    being used are not cryptographic related :-).
 * 4. If you include any Windows specific code (or a derivative thereof) from
 *    the apps directory (application code) you must include an acknowledgement:
 *    "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
 *
 * THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * The licence and distribution terms for any publically available version or
 * derivative of this code cannot be changed.  i.e. this code cannot simply be
 * copied and put under another distribution licence
 * [including the GNU Public Licence.]
 */
```

10.23 libssh2

```
/* Copyright (c) 2004-2007 Sara Golemon <sarag@libssh2.org>
 * Copyright (c) 2005,2006 Mikhail Gusarov <dottedmag@dottedmag.net>
 * Copyright (c) 2006-2007 The Written Word, Inc.
 * Copyright (c) 2007 Eli Fant <elifantu@mail.ru>
 * Copyright (c) 2009 Daniel Stenberg
 * Copyright (C) 2008, 2009 Simon Josefsson
 * All rights reserved.
```

```
 *
 * Redistribution and use in source and binary forms,
 * with or without modification, are permitted provided
 * that the following conditions are met:
 *
 *   Redistributions of source code must retain the above
 *   copyright notice, this list of conditions and the
 *   following disclaimer.
 *
 *   Redistributions in binary form must reproduce the above
 *   copyright notice, this list of conditions and the following
 *   disclaimer in the documentation and/or other materials
 *   provided with the distribution.
 *
 *   Neither the name of the copyright holder nor the names
 *   of any other contributors may be used to endorse or
 *   promote products derived from this software without
 *   specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
 * CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
 * INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
 * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
 * CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 * SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
 * BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
 * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
 * WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
 * NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE
 * USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
 * OF SUCH DAMAGE.
 */
```

Symbols

3D data hull, 80
3D dataset, 13
    configuration, 19
    editiong isolines, 17
    inserting isolines, 18
    inserting isolines labels, 19
3D diagram, 32

A

Acrobat-Reader, 5
arrow, 39
arrow object, 34
ASAM-ODS, 65
axes
    hide, 26
    link, 26
    log scale, 26
    move, 26
    position, 26
    time/date scale, 26
axis title
    arrow, 39
    insert marker, 39

C

CAS, 65
circle object, 34
clipboard, 51
COMBI, 65
copy, 51
copy page, 51
Create your own text Legends, 48
CSV-Import, 70
cut, 51
cxImage, 103

D

data import, 13, 65
datasets
    layering, 25
dBase, 65
DIADem, 65
diagram
    2D-Diagram, 26
    3D diagram, 32
    axis position, 26
    data import, 13
    hide axis, 26
    layering, 25
    link axis, 26
    move axis, 26
    setup, 26
    Waterfall, 33
drawing objects
    arrow, 34
    circle, 34
    editing, 36
    ellipse, 34
    Formulas, 36
    grouping, 36
    layering, 25, 36
    line, 34
    moving, 34
    rectangle, 34
    resizing, 34
    selecting, 34
    text, 34, 36
Drawing Toolbar, 34
DSV-Import, 70

E

Ellipse, 34

Excel File Import, 70

F

Famos, 65
FEVIS, 65
Field Functions, 44
    Filename, 44
field functions
    upper/lower case, 44

G

Greek letters, 37

H

header, 47
http://www.uniplot.de, 5
hull
    load, 83
    save, 83

I

IFILE, 65
import
    data, 13
    options, 14
INCA, 65
INDIMASTER, 65
INDISET, 65
info box, 47
Installation, 1
Internet, 5

J

jpeg, 101

L

Legend, 44
legend, 39, 40, 47
libxml2, 107
License, 94
line, 38
line object, 34
load hull, 83
Lotus, 65
LZ4, 107
lzf, 99

M

marker, 39
MDF, 65
Mouse wheel, 35

N

netCDF, 15, 65, 95

O

OLE, 51
OnLayerLink, 29

P

Page, 24, 25
    Drawing Order, 26
page import, 47
Page Margin, 25
paste, 51
PCRE-Lizenz, 97
Placeholder ($Text$), 42
png, 101

Q

qhull, 100

R

rectangle object, 34
replot, 18

S

save hull, 83
Scintilla, 96
Setup, 1
Special Characters, 37
stacked diagrams, 29
symbol, 39

T

Table Object
    Field Functions, 44
Tabs in Text objects, 41
Text File Import, 70
Text Object
    Field Functions, 44
text object, 34, 36
    arrow, 39
    greek letters, 37
    line, 38
    marker, 39
    spaces in text, 40

special characters, 37
subscript, 36
superscript, 37
Tabs, 41
update size automatically, 44
Text Placeholder ($Text$), 42
Toolbar
Cursor, 10
Drawing, 34

U

Updates, 5
UTX, 65
UTX file format, 71
UXX-BEGIN keyword, 71
UXX-Import, 78

V

VS100, 65
VTK, 97

W

WFT, 65
WOT curve, 80

X

xBase, 65

Z

zlib, 99
zoom, 35